# Exhibit A

**UNITED STATES DISTRICT COURT**
**FOR THE DISTRICT OF MASSACHUSETTS**

| | |
|---|---|
| **ANYWHERECOMMERCE, INC. and BBPOS LIMITED,**<br><br>**Plaintiffs,**<br><br>**v.**<br><br>**INGENICO, INC., INGENICO CORP., INGENICO GROUP, SA, and INGENICO VENURES SAS,**<br><br>**Defendants.** | **Civil Docket No: 1:19-cv-11457-IT** |

# EXPERT REPORT OF IVAN ZATKOVICH

**FEBRUARY 16, 2022**

**CONFIDENTIAL**

# TABLE OF CONTENTS

# LIST OF EXHIBITS

| Exhibit No. | Description |
|---|---|
| **A** | Curriculum Vitae of Ivan Zatkovich |
| **B** | List of Materials Considered |

# 1 INTRODUCTION

1. I have been retained by Plaintiff BBPOS Limited ("BBPOS"), through its attorneys, Kutak Rock LLP, as an expert in this case. Among other things, I have been asked to evaluate the evidence and testimony provided to me, along with the results of my in-depth research, to provide an opinion on defendants' use of BBPOS' proprietary information based on my extensive background and expertise in the subject matter. Additionally, I have been asked to provide background information relative to certain technical issues as set forth herein.

2. This report sets forth the opinions I have formed in this matter and the bases for those opinions through my independent evaluation and analysis. My opinions are based on the information available to me as of the date that I am submitting this report. If additional information becomes available to me either by production by the parties or third parties, or otherwise, I may, if permitted to do so, offer additional opinions. I may also, if requested and permitted to do so, provide further opinions to rebut any testimony, reports or opinions offered by Defendant's witnesses (expert or otherwise). I may present demonstrative or illustrative exhibits at trial to explain my opinions.

## 1.1 Report Objectives

3. This section describes what I have been asked to perform by Kutak Rock LLP, in the case of *AnywhereCommerce, Inc., et al. v. Ingenico, Inc., et al*., pending in the United States District Court for the District of Massachusetts, Civil Docket No: 1:19-cv-11457-IT. I have been asked to evaluate and opine on the following areas:

- Identify what, if any, proprietary information BBPOS has created in the development of their products. (Section 5 – BBPOS trade secrets)

- Identify what BBPOS proprietary information that Ingenico obtained. (Section 6)
- Determine what BBPOS proprietary information, if any, Ingenico misappropriated (Section 7)

4. As part of my research and analysis in this matter I have referenced the documents listed in Exhibit 2 – Material Considered.

### 1.2 Summary of Preparer's Qualifications

5. Listed below is a brief summary of my qualifications. A more detailed description of my background appears in Exhibit A CV of Ivan Zatkovich.

6. I am an eBusiness Director and Technology Consultant specializing in Telecommunications, eCommerce, Billing and Payment systems. I have been an industry speaker for software development practices and technology integration. I have over 30 years' experience in a diverse set of technologies including wireless communications (Wi-Fi, Bluetooth, GSM, and broadband), credit card payment processing, credit card payment processing, secure payment transactions for EFT, POS, payment gateways, and mobile payments, operation of POS & mPOS payment card devices including magnetic stripe, EMV smart-chip, contactless NFC, and magnetic secure transmission (MST).

7. The following are examples of my professional work experience:

- **Evatone** - Developed multiple credit card payment methods and payment gateway solutions for eCommerce clients such as Warner Brothers, Wachovia Bank, and Pro Marine.

- **eComp Consultants** - developed patent claims to cover secure mobile payment transactions and account access using user id authentication.
- **Verizon** – Telecommunications - Implemented applications for telephone networks, wireless data communications using 802.11 and Bluetooth on wireless, and local secure networks. As well as audio and data encryption for VoIP, fax and modem application. Developed Data Transaction Authentication and verification (to identify tampering, or modification).
- **Tanning Technology** - System Architect and developer of financial systems for:
  - ETrade Online Trading – Developed 1st version of Securities trading system
  - Smith Barney - PDA Trader Access - Wireless retrieval of financial transactions information
  - GEICO, Hartford insurance – secure online systems for payment transactions and policyholder claims processing

8. I am a frequent industry speaker/presenter and, perhaps of particular relevance, a certified eCommerce solutions expert. I have also been a testifying expert for over 60 cases including patent litigation, ITC, IPRs, and CBMs for secure financial transactions, point-of-sale, mobile and wireless payment authentication.

- **Plaintiff v Samsung – ITC** - Retained as testifying expert for mobile device technology for emulation and transmission of MST (Magnetic Secure Transmissions) transactions to Magnetic Stripe card reader terminals.

- **Black Hills v Sonos – Patent Litigation** - Testifying expert for analyzing multiple audio sources, Bluetooth and mesh network protocols, and synchronized digital audio signals.
- **3M Futures South Africa v. Standard Bank - Patent Litigation** - Real-time credit card transaction authorization, Mobile Payment authorization, and User Authentication. Testified at Trial regarding Invalidity and Infringement in South Africa.
- **Alexsam v. MasterCard – Patent Litigation** - Testifying expert for Multifunction credit card systems. Technology covered activation and processing of multiple transaction types from point of sale, through Credit Card network and processing hub, to Issuing Banks. Including security and tokenization features, processing points, and routing of transactions.
- **Paul Ware v Aldo Group, Inc, et al. - Patent Litigation** - Providing expertise in eCommerce, Payment Gateways, Point-of-Sale user interfaces, specifically credit card readers, and Credit Card transaction processing.
- **IslandIP, Inc. v. Deustche Bank - Patent Litigation** – Testifying expert for bank transactions for Sweep Accounts, including features for interest bearing, FDIC insured, and Omnibus deposits and withdrawals.
- **Askeladden v Smart Verify, N5 – IPRs** – Testifying expert for Multi-factor User Authentication for card transactions.

### 1.3 Compensation

9. I am being compensated as an independent consultant in this matter at a rate of $525 per hour for my work on this matter. I have received no additional compensation for my work on this matter, and my compensation depends in no way on my opinions expressed in this report, any testimony that I may give, or the outcome of this matter. In addition, I will be reimbursed for reasonable expenses incurred in connection with my work on this matter.

### 1.4 Materials Considered

10. In preparing my expert report, I have relied upon my knowledge, skill, experience, training, and education in my field of expertise. I have also reviewed and considered:

- BBPOS Hardware, Software & Firmware
- Ingenico Devices and Software
- Deposition testimony and deposition exhibits for Defendant's witnesses
- documents produced by both parties and third parties

11. A list of materials that I have relied upon in forming the opinions set forth in this report is attached to this report as Exhibit B, as well as cited throughout this report and in the accompanying exhibits. I reserve the right to consider and rely on, as well as supplement, this report in view of additional information that is provided to me in this review, including information considered by Defendant's experts or developed before trial.

## 2 BASIC LEGAL PRINCIPLES

12. I have been provided certain legal principles and standards by counsel, which are shown below, relating to the issues in this report. In conducting my analysis set forth in this report, I have been guided by these principles.

| | |
|---|---|
| Most Pertinent Legal Principles | In Count II of its First Amended Complaint, BBPOS asserts a claim for violation of the Georgia Trade Secrets Act against the defendants. This claim is subject to a five (5) year statute of limitations. O.C.G.A. § 10-1-766 provides:<br><br>> An action for misappropriation must be brought within five years after the misappropriation is discovered or by the exercise of reasonable diligence should have been discovered. For the purposes of this Code section, a continuing misappropriation by any person constitutes a single claim against that person, but this Code section shall be applied separately to the claim against each person who receives a trade secret from another person who misappropriated that trade secret.<br><br>In Count III, BBPOS asserts a claim for violation of the Massachusetts Trade Secrets Act against the defendants. This claim is subject to a three (3) year statute of limitations. M.G.L.A. ch. 93, § 42E ("An action for misappropriation must be brought within 3 years after the misappropriation is discovered or by the exercise of reasonable diligence should have been discovered. For the purposes of sections 42 to 42G, inclusive, a continuing disclosure or use constitutes a single claim.").<br><br>In Count IV, BBPOS asserts a claim for violation of the Defend Trade Secrets Act of 2016 against the defendants. 18 U.S.C.A. § 1836(d), Period of limitations, states:<br><br>> A civil action under subsection (b) may not be commenced later than 3 years after the date on which the misappropriation with respect to which the action would relate is discovered or by the exercise of reasonable diligence should have been discovered. For purposes of this subsection, a continuing misappropriation constitutes a single claim of misappropriation.<br><br>Relevant to all three (3) claims:<br><br>> To prevail on a claim for violations of the trade secret theft statutes, a plaintiff must prove four elements: (1) that a plaintiff's alleged trade secrets meet the statutory definition of trade secrets; (2) that the defendant misappropriated those trade secrets; (3) that the plaintiff has been actually harmed as a direct result of the misappropriation; and (4) that the alleged trade secrets are related to a product or service used in, or intended for use in, interstate or foreign commerce.<br><br>There are six ways that a person or an entity can be liable for misappropriating a trade secret:<br><br>> (1) by acquiring the trade secret of another while knowing or having reason to know it was improperly acquired;<br>(2) by disclosing or using another's trade secret, without express or implied consent, after acquiring knowledge of it by improper means; |

| | |
|---|---|
| | (3) by disclosing or using another's trade secret, without express or implied consent, knowing or having reason to know at the time of disclosure or use that knowledge of the trade secret had been derived from or through a person who had used improper means to acquire it;<br>(4) by disclosing or using another's trade secret, without express or implied consent, knowing or having reason to know at the time of disclosure or use that knowledge of the trade secret had been acquired under circumstances giving rise to a duty to maintain its secrecy or limit its use;<br>(5) by disclosing or using another's trade secret, without express or implied consent, knowing or having reason to know at the time of the disclosure or use that knowledge of the trade secret was derived from or through a person who owed a duty to the person seeking relief to maintain its secrecy or limit its use; or<br>(6) by disclosing or using another's trade secret, without express or implied consent, knowing or having reason to know, before a material change in position, that knowledge of the trade secret had been acquired by accident or mistake.<br><br>O.C.G.A. § 10-1-761(2).<br><br>"Person" is broadly defined in the Georgia Trade Secrets Act. It includes a natural person, corporation, business trust, estate, trust, partnership, association, joint venture, government, governmental agency or subdivision, or any other profit or nonprofit legal or commercial entity. O.C.G.A. § 10-1-761(3).<br><br>"Improper means" include theft, bribery, misrepresentation, breach or inducement of a breach of a confidential relationship or other duty to maintain secrecy or limit use, or espionage through electronic or other means. O.C.G.A. § 10-1-761(1).<br><br>Misappropriation arises when one who has legitimate access to trade secret information, such as in the course of his or her employment or by virtue of a license or other type of agreement (*CMAX/Cleveland, Inc. v. UCR, Inc.*, 804 F. Supp. 337 (M.D. Ga. 1992) (applying Georgia law)), takes advantage of that knowledge by using it to develop or market his or her own product (*Specialty Chemicals & Services, Inc. v. Chandler*, 9 U.S.P.Q.2d 1793, 1988 WL 618583 (N.D. Ga. 1988) (applying Georgia law); *Thomas v. Best Mfg. Corp., Division of Tillotson Corp.*, 234 Ga. 787, 218 S.E.2d 68 (1975)) or discloses the product to unauthorized persons. *Georgia-Pacific Corp. v. Lieberam*, 959 F.2d 901 (11th Cir. 1992) (applying Georgia law).<br><br>This is true even if the final product has been modified as long as the product is substantially derived from the original. *EarthCam, Inc. v. OxBlue Corp.*, 49 F. Supp. 3d 1210 (N.D. Ga. 2014), aff'd, 703 Fed. Appx. 803 (11th Cir. 2017) (applying Georgia law) (further providing that if the contribution made by |

| | |
|---|---|
| | the trade secret is so slight that the actor's product or process can be said to derive from other sources of information or from independent creation, the trade secret has not been "used" for purposes of imposing liability); *Specialty Chemicals & Services, Inc. v. Chandler*, 9 U.S.P.Q.2d 1793, 1988 WL 618583 (N.D. Ga. 1988) (applying Georgia law).<br><br>However, a licensee may use trade secrets as long as the use is within the scope of the license agreement. *Philip Morris, Inc. v. Brown & Williamson Tobacco Corp.*, 641 F. Supp. 1438 (M.D. Ga. 1986), order clarified, 645 F. Supp. 174 (M.D. Ga. 1986) (applying Georgia law). |

# 3 CASE HISTORY / BACKGROUND

## 3.1 The Parties

### 3.1.1 BBPOS

13. BBPOS specializes in Mobile Point of Sale devices (mPOS). In fact, that is the reason for the founding of the company. The founders had the idea of providing a communication method for POS transactions using a cell phone coupled with an attached POS device. This device would be fully contained as a single unit and portable, or mobile, just like a cell phone. The mPOS unit BBPOS designed would attach to the audio jack of a cell phone to receive power and provide communications to a cell phone for transmitting the transaction to the payment network to complete the process.

14. BBPOS began development of their mPOS devices in 2008 and formed the company that same year. Their mPOS devices were originally designed for audio communications but evolved into using USB and Bluetooth connections as well. All devices developed by BBPOS have been mPOS devices. BBPOS began their first shipments into North America in 2008, first to Canada and then in 2011, BBPOS was shipping 400K units annually to all of North America. By 2012, BBPOS cleared shipping 1MM units that year and continue to grow. Initially in 2008, only 2 companies had developed mPOS devices; BBPOS was one of them and the only one with

encryption of the data transferred between the reader and the mobile phone. BBPOS is clearly a leader in mPOS technology.

15. Prior to engaging with Defendant ROAM Data, Inc. ("ROAM"), BBPOS had developed their own suite of mPOS products, specifically the G3X which was developed prior to Sept 2010 and formed the basis for the agreement with ROAM. BBPOS was actively developing, producing and certifying these devices along with its associated SDK (Software Developer Kit), advanced security methods, innovative hardware design, communication protocols, and APIs. At the time of their agreement with ROAM in May 2010, BBPOS was already known as a leader in mPOS technology.

16. The original mPOS devices were Magstripe Swiper devices for use with Magstripe cards. BBPOS also has several issued patents that disclose unique technology specifically related to mPOS devices dating back to 2009.

17. Plaintiff AnywhereCommerce, Inc. ("AnywhereCommerce") is a d/b/a affiliate of non-party, 4361423 Canada Inc. ("436 Canada"), which is the assignee of record and owner of certain patented technology invented by the principals of BBPOS, among others, and is in the business of making, using, or selling credit card readers.

18. HomeATM ePayment Solutions, Inc., a Canadian corporation that conducted business under various trade names, including "HomeATM" and "HomeATM ePayment Solutions" ("HomeATM"), is a predecessor in interest to certain of 436 Canada's mPOS patent rights, among other things, and certain of AnywhereCommerce's business operations, via an amalgamation transaction pursuant to the Canada Business Corporations Act.

19. Under a License Agreement and Non-Competition Agreement dated March 18, 2010 (the "HomeATM-BBPOS License"), by and between HomeATM and BBPOS, HomeATM

granted to BBPOS "**a non-exclusive licence** [sic] to use the Intellectual Property Rights in the Region in relation to the Products." *See* HomeATM-BBPOS License at §2.1. The scope of the licensed rights granted reads as follows:

> *(1) "Intellectual Property Rights" is defined as "trademark, design, or patent and other intellectual property rights of* ***DTMF technology titled the Apparatus and Method for Commercial Transactions Using a Communication Device****".*
>
> *See §1.1 (emphasis in original).*
>
> *(2) "Products" is defined as "the communication device using the DTMF technology".*
>
> *Id. The rights and benefits arising under the contract were not assignable to any unrelated third party "without the prior written consent of the other party [thereto]." Id. at 9.1.*
>
> ***HomeATM-BBPOS License Agreement [AC_0000917-922]***

20. Defendant Ingenico Group S.A. ("Ingenico") was incorporated in France in 1980 and sells payment solutions. Ingenico Corp. and Ingenico Inc. are Ingenico Group S.A.'s subsidiaries and were incorporated in the United States. *See* Def. Answer, ¶¶1, 39.

21. In November 2009, Ingenico had acquired a minority ownership interest in ROAM Data, Inc. ("ROAM").[1]

22. At the beginning of 2010, ROAM had a software-based payment card solutions platform and plans for developing an acoustic-based mPOS device device (e.g. audio jack interface), but had not yet entered the mPOS device sales market, which was quickly about to

[1] ROAM is the name of the company that formerly participated in the business of mobile payments. On or around February 6, 2012, Ingenico held more than 70% of the ownership interests in ROAM. Def. Answer, ¶44. It acquired 100% of the ownership interests in ROAM by January 1, 2015. Def. Answer, ¶¶8, 53. On December 31, 2017, ROAM merged with and into Ingenico. Def. Am. Counterclaims, ¶12.

accelerate. Partnering, or acquiring the mPOS technology would clearly produce a quicker market position for ROAM in this arena.

23. On May 4, 2010, BBPOS and Ingenico's predecessor-by-merger, ROAM, entered into an exclusive product license, as evidenced, in part, by the Engineering Development and License Agreement as amended on August 15, 2011 (the "BBPOS-ROAM Licensing Agreement").

### 3.1.2 ROAM

24. ROAM Data received their Series A funding in Jan 2008 and a Series B funding in Nov 2009, whereafter, Ingenico owned a minority interest in the company.[2] The ROAM Data flagship product at that time was a software-based payment and merchant services platform. ROAM Data had not invested in development of an mPOS device and as the market demand was increasing in this area, they looked to invest in a partner who could speed their time to market for such an offering.

25. Through an introduction by HomeATM, ROAM was introduced to BBPOS in early 2010. This introduction led to the inking of the BBPOS-ROAM Licensing Agreement.

26. Upon the agreement between BBPOS and ROAM Data; ROAM began marketing to players interested in the mPOS technology. For example, in January of 2012, ROAM signed an MNDA with PayPal to begin discussions about producing a PayPal specific model of mPOS. By May 2012, PayPal and ROAM Data reach an agreement to produce the PayPal model based on the G3X evolution. PayPal began testing with the G4X product prior to creating their own physical form factor, the blue triangle, which then became the G5X.

[2] As noted above, ROAM received additional investments from Ingenico in Feb 2012 and later was fully acquired by Ingenico in January 2015.

### 3.1.3 Ingenico

27. Ingenico is a leader in terminal-based Point of Sale devices and has been developing them since 1984. Ingenico invested in ROAM and Fujian Landi Commercial Equipment Co Ltd ("Landi"), a Chinese manufacturing company, to expand their reach in emerging POS technologies, handhelds, wireless and eventually mobile.

28. Prior to engaging with BBPOS / ROAM, Ingenico had not developed any mPOS products. While mPOS was listed as a future product, their R&D group had not set any designs in motion as of Dec'11.

29. Ingenico designated their product portfolios based on device types. It is helpful to understand the product families and classifications when reviewing their product information:

- iWB - Wireless Biometric Terminal
- iWL = Wireless
- iCT = Countertop Terminal
- iCMP - Companion Mobile POS
- iTMP - Thin Mobile POS

30. As part of the iTMP portfolio, Ingenico and Landi produced the following models of mPOS devices.

| Model | Function | Timing | Development Notes |
|---|---|---|---|
| RP100x | Swipe Only/PayPal | 2013 | Roam Data/Ingenico with Landi<br>- based on the R350x platform |
| RP150x | Swipe Only | 2013 | Roam Data/Ingenico with Landi<br>- based on the R350x platform |
| RP350x | Chip & Swipe | 2013 | Roam Data/Ingenico with Landi |
| RP750x | Chip & PIN, Contactless | 2013 | Roam Data/Ingenico with Landi<br>- based on the R350x platform |
| RP170c | Magstripe & Contactless Reader | 2014-2016 | Roam Data/Ingenico with Landi<br>- based on the RP750x platform<br>- no chip reader<br>- came just before the RP457c |

| | | | |
|---|---|---|---|
| RP450 | Magstripe, EMV Chip & Contactless Reader | 2015+ | Roam Data/Ingenico with Landi<br>- based on the RP750x platform<br>- includes audio jack |
| RP456 | Magstripe, EMV Chip & Contactless Reader | 2015+ | Roam Data/Ingenico with Landi<br>- based on the RP750x platform<br>- includes audio jack and Bluetooth (BT) |
| RP457 | Magstripe, EMV Chip & Contactless Reader | 2015+ | Roam Data/Ingenico with Landi<br>- based on the RP750x platform<br>- includes audio jack, Bluetooth (BT) and MFI |

### 3.1.4 Landi

31. Landi had been developing POS devices since 2006. Landi was acquired by Ingenico in 2012. Landi, as a wholly owned subsidiary of Ingenico, markets their devices separately as Landicorp and as part of the Ingenico portfolio.

32. Prior to BBPOS's joint development project, Landi had developed many varieties of POS products. However, none of them had been Mobile POS (mPOS) products using an Audio Connection for communication between a mPOS device and a cell phone. The only mention of a Landi developed mPOS device appears in Ingenico/Landi product roadmaps from late October, 2011 which show a possible 3Q2012 launch. More detailed roadmap slides from the same presentation show the mPOS device as having no funding nor milestone projections. In my experience, implementing a product in Q2 with no funding nor project plan as of late the prior year would be virtually impossible without acquiring the desired technology already in process from a third party.

## 3.2 The BBPOS-ROAM Licensing Agreement between BBPOS and ROAM

33. BBPOS-ROAM Licensing Agreement, the operative agreement at issue in the lawsuit between BBPOS and ROAM, executed in May'10, contemplated for BBPOS to develop

and provide Mobile POS devices to ROAM Data and to provide associated product software, solutions, support and services. The agreement includes clauses to protect the intellectual property of both parties and clearly states that the agreement is non-transferrable should ROAM be sold to a competitor.

34. The agreement also allowed for BBPOS to continue to sell their products in China and the Philippines.

35. The BBPOS-ROAM Licensing Agreement was amended in Aug'11 to address patents filed by BBPOS during their continued development of the emerging mPOS technology. The amendment also includes new compensation in response to the newly developed IP.

36. This agreement was signed by Will Graylin, CEO ROAM, and Ben Lo, CEO BBPOS. My understanding of this agreement is that it provides for the following terms and conditions.

| | |
|---|---|
| **Engineering Development and License Agreement**<br><br>- Between ROAM Data, Inc and BBPOS Limited<br>- Executed May 4, 2010<br>- Amended August 15, 2011 inc. patent filing 12/767,831 and new Schedule II - Compensation<br><br>**Schedule I – Products, Solutions, Devices, Services & Specifications**<br><br><u>Products</u><br>- Encrypted CircleSwipe Reader, aka "Crypto Swipe" or "ROAMpay Swipe"<br>- EMV capable POS unit with Bluetooth interface, aka the "BBPOS"<br><u>Solutions</u> | - Grants ROAM exclusive rights to use and sell products developed by BBPOS worldwide except for China and Philippines, for which BBPOS retains nonexclusive rights.<br>- The license is not transferrable or assignable should ROAM Data be sold to a competitor with its own POS products.<br>- Includes nonexclusive rights for BBPOS to resell the BBPOS and ROAMpay POS solution and will be compensated recurring revenue associated with the resale transaction.<br>- The agreement covers any future SOWs for Products, Devices, Deliverables, Services or Specifications.<br>- "6.1 Both parties agree to treat the other party's Confidential Information as confidential (as defined below), to take all reasonable measures to protect and prevent the disclosure of and/or unauthorized use by third parties of the other parties' Confidential Information, to exercise at least the same degree of care exercised for |

| | |
|---|---|
| - Payment Solution using the "BBPOS" or "ROAMpay POS" … transact via ROAM data gateway or equivalent<br>Devices<br>- The "ROAMpay POS" EMV terminal<br>Services and Specifications<br>- Appropriate design and manufacturing services to produce the Crypto Swipe devices, including key injection and support to port ROAMplayers top a variety of devices.<br>Software Development Services<br>- Design and development of ROAMpay POS device, incl, EMV L1, L2, and PCI certifications. Future versions may contain NFC. | the production of its own confidential Information, and to use Confidential Information other than for its intended purpose under this agreement."<br>- "6.3 Notwithstanding any provision herein to the contrary, a party may disclose Confidential Information on a need-to-know basis to its contractors, lawyers, accountants and agents, provided that any such person is bound by a duty of confidentiality and such party shall be responsible for any disclosure or use by any such third party in contravention hereof. . ."<br>- "6.5 This obligation not to disclose shall continue for a period of five (5) years after the termination of this Agreement." |

***BBPOS ROAM Engineering and License Agreement [AC_02770544-554, IngenicoInc_0268234-238]***

### 3.3 Summary of activities between BBPOS and ROAM Data

37. From Feb – July of 2012, BBPOS and ROAM were heavily engaged in projects to build and certify several versions of a mPOS audio connected device. This was in part to support the efforts of PayPal to create a new offering for their customers. As such, ROAM and PayPal signed a mutual NDA in Jan'12 followed by a formal engagement agreement in May'12.

38. The engagement between ROAM Data and BBPOS included jointly developing the following products:

- G4X developed as a general purpose mPOS for ROAM but used as a testing platform for PayPal
- G5X (G4X function in PayPal form factor)

39. Ingenico who had invested in ROAM Data, was also very interested in the mPOS development and inserted a new VP of mPOS Product Management, Christopher Rotsaert, to

oversee/manage the mPOS development effort. Immediately prior, Mr. Rotsaert had overseen and managed Ingenico's recent acquisition of Landi as Ingenico's China Operations Manager. Ingenico's reasoning was to have oversight of the project and be able to provide assistance for quicker speed to market since Ingenico was a leader in POS systems and had extended development arms (Ingenico's Valence development team & Landi) to leverage as extra resources. Mr. Rotsaert's new position with ROAM was funded jointly by ROAM and Ingenico.

40. During this time BBPOS shared much of their mPOS product development knowledge and trade secrets with ROAM Data, including hardware designs and schematics, software designs and source code, and various methods for encrypting and communicating credit card information. This included sharing this same data with the joint Product Manager.

41. The joint Product Manager, Christopher Rotsaert, had a broad reach in the Ingenico group of companies which included Landi. So, while ROAM was sharing BBPOS confidential information with the Ingenico Product Manager, subject to the contractual protections set forth in the BBPOS-ROAM Licensing Agreement at section 6.3, and elsewhere, , he was, in turn, also sharing it with Landi, without BBPOS's knowledge or approval. This joint Product Manager also established a communication link between the ROAM Data Dev team and the Landi Dev Team in order to execute joint projects such as trade show and customer demos.

42. BBPOS and ROAM Data were operating under contract terms which protected the BBPOS trade secrets from third parties, even those acquiring ROAM Data.

43. As noted above, Ingenico had made an investment in ROAM as of 2009 and increased its investment to hold a majority stake in ROAM in February 2012, ultimately fully acquiring the company in 2015.

44. The BBPOS, ROAM Data, and Ingenico's objectives were to produce products for PayPal (among many other potential clients, e.g. Capital One).

### 3.4 Proposed Acquisition of BBPOS by ROAM Data / Ingenico

45. BBPOS was very important to ROAM Data's product strategy and therefore ROAM approached BBPOS to acquire them and their associated IP. A discussion and original term sheet was developed in Sept'11. BBPOS, having had a good working relationship with ROAM Data entertained acquisition discussions and after several iterations they had come to non-binding terms in Mar'12. ROAM Data informed BBPOS that the acquisition deal had to pass thru Ingenico Due Diligence in order to close; the Ingenico Due Diligence began in Apr'12. BBPOS supplied many documents including trade secret information with Ingenico as part of the Due Diligence.

### 3.5 Abandonment of Acquisition Talks

46. During the course of Ingenico's Due Diligence, BBPOS received a large purchase order in the Chinese market in or around June 2012, which upset the economics of the deal.

47. When BBPOS sought a carve-out for the Chinese market, Ingenico was incensed and threatened to destroy BBPOS if it did not go through under the original (non-binding) terms of the deal. Ingenico's CEO, Phillipe Lazare, flew to Hong Kong to deliver this message to BBPOS, personally. The negotiations then abruptly were terminated.

### 3.6 Ingenico Request for Confidential Information

48. Meanwhile, under the protections of ROAM's contractual confidentiality obligations set forth in the BBPOS-ROAM Licensing Agreement, BBPOS continued in its efforts to deepen its growing business relationship with ROAM.

49. By Jul'12, the Ingenico PM begins requesting information from the BBPOS team and disseminates it to the Ingenico Dev teams. The first information obtained is regarding the audio jack sampling rate, AGC and power management.

50. For example, from July 15-27, 2012, the Ingenico Product Manager, Christopher Rotsaert, requests and obtains the following proprietary information and designs from the BBPOS team for the purposes of preparing for introduction to the German Market as well as the Press Release in Cartes Paris (Nov'12) including:

- Battery life estimation
- BBPOS Data Communication / Output Formats (V21)
- Supported list of Mobile Phones and attributes
- Audio interface specifications
- PayPal device schematics
- Audio Interface design schematics
- Swiper Track 2&3 formats
- Power consumption and management for coin battery devices vs battery-less devices and how power is handled for low power devices needing battery compensating designs.

### 3.7 ROAM – BBPOS relationship change

51. At the end of Aug'12, Will Graylin, ROAM CEO and Christopher Rotsaert, Product Manager have a disagreement over terms of Mr. Rotsaert's unauthorized sharing of BBPOS' confidential information while it was under contract to protect same.

52. In an email dated September 17, 2012 directed to Christopher Rotsaert, Ingenico's appointee to ROAM, Mr. Graylin observed "[a]pparently you are still not comprehending the gravity of the situation[,]" outraged by Ingenico's actions relative to BBPOS's trade secrets:

> *"Just because you sent me an email to me, does not mean you have my agreement and my permission to start* ***transferring IP that does not belong to Ingenico****.* ***Your assumption that the reader IP belongs to ROAM was already incorrect****, then to further transfer them further to Ingenico without my explicit permission and without any commercial agreement in place was a* ***real mistake****. Apparently you are still not comprehending the gravity*

> *of the situation. Your actions and assumptions are threatening the very fabric of ROAM's relationship with its most important supplier. You are transferring value from one company to another company unilaterally without agreement or consideration.* ***There is an apparent lack of respect for the IP or BBPOS and ROAM****, and for my role as CEO of ROAM."*

***See Email Thread dated August 29, 2012 – September 17, 2012 by and between Will Graylin and Christopher Rotsaert (emphasis added).***

53. By early Sept'12 the Ingenico Dev team appears to be trying to figure out the BBPOS solution on their own with limited success. [IngenicoInc_0283891-IngenicoInc_0283892]

54. By late Sept'12, the decision is made by the Ingenico PM to stop development with BBPOS on iTMP[3] and move to product development with Landi. [IngenicoInc_0283893] As such, a SOW is created between Ingenico and Landi for the iTMP development [IngenicoInc_0138722-IngenicoInc_0138748]

55. The ROAM CEO is replaced in Oct'12.

## 4 TECHNICAL BACKGROUND - POS AND MPOS DEVICES

### 4.1 Traditional POS devices

56. POS stands for Point of Sale. POS devices, specifically POS devices that read cards such as debit or credit cards, include devices that range from large cash registers with card readers to handheld POS devices that a waitress can bring to your table to take your payment.

[3] The iTMP platform refers to "Thin Mobile POS" product line which is being marketed to PayPal, etc.


**POS devices**

***Cash Register w/ card reader*** ***Handheld card reader***

The primary components of these POS devices, specifically regarding accepting payment cards, is:

- **The card reader –** Reads cards by swiping the magstripe, inserting the smart chip, or wireless "tap and pay" using NFC protocols (Near Field Communications).
- **Transaction communication –** this is the ability to transmit the card data and/or payment transactions from the POS device to a payment service or payment gateway through a network. For cash registers, this may be done with wired networks. For Handheld card readers this may be done over WIFI.

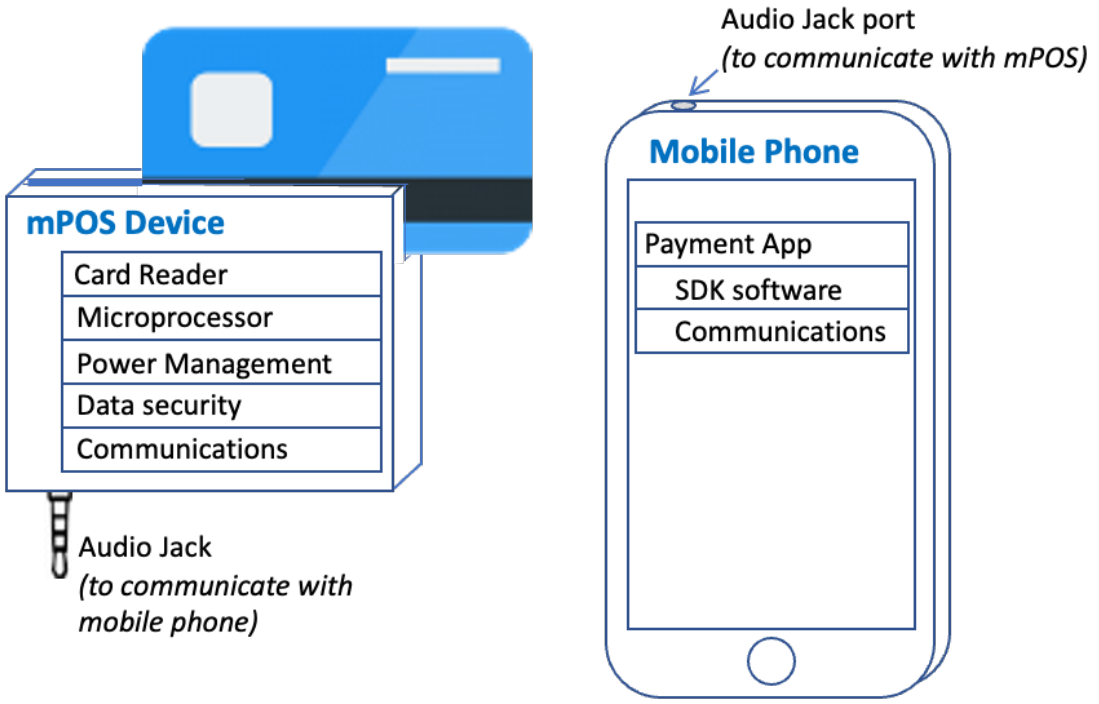
### 4.2 mPOS Devices

57. Unlike traditional POS devices, an mPOS device that a merchant uses to accept card payments (circa 2012) is a specific type of POS device that is typically connected to a mobile phone. These devices need to be smaller (and usually "smarter") than typical POS devices since

they have to communicate securely with many types of mobile phones and manage the power of their smaller batteries more efficiently.


***Typical mPOS device, 2012***

58. mPOS devices require additional components and technology than the traditional POS devices. The following shows the primary components and technology of an mPOS device as well as the Mobile Payment application within the Mobile Phone that it must communicate with.




*Components of an mPOS device with Mobile Phone application*

59. The components to developed for an mPOS device include:

a. **Card Reader –** can include a magnetic card swiper reader, and smart-chip card reader, and/or a wireless 'tap to pay' NFC card reader.

b. **Microprocessor –** this, along with software / firmware on the device, provides the controller to perform card transactions.

c. **Power Management –** because of the limited battery size in typical devices such as this, efficient use of power is essential. This would include circuitry and software.

d. **Data Security –** this includes industry standard methods for retrieving secure data from payment cards as well as possible proprietary methods to encrypting data to be sent to/from the mobile phone.

e. **Communications** – This includes specific data formats and methods used to communicate between the mPOS device and the mobile application on the Mobile phone.

f. **Audio Jack** – using an audio jack to communicate data to a mobile phone was typical for mPOS devices in 2012 since many phones did not support interfaces such as Bluetooth. This required converting the digital card data into audio signals to send the data over the audio jack, and then converting it back to digital data on the receiving end.

60. The Payment application on the Mobile Phone is provided to the Merchant to accept and process payment transactions by communicating with the mPOS device to retrieve the card data. The mobile phone payment application is typically provided by a payment processor such as Square, PayPal, and Bank of America. The components of the Mobile Phone and Payment app includes:

a. **SDK software –** (Software Development Kit) this is a library of functions provided by the mPOS manufacturer to the Payment processors to allow the payment applications to operate with the mPOS device.

b. **Communications** – this includes software to communicate with the mPOS as well as communicate with the Payment Gateway or Server to process the card transaction with the Merchant bank and Card issuing bank.

c. **Audio Jack Port** – this provides the typical method to send and receive the data from the mPOS device as audio signals.

61. With the exception of the Card Reader itself and some aspects of the Microprocessor use, the manufacturers of traditional POS devices do not have designs nor development processes in place to create the other components and technology that is required for mPOS devices. Therefore, new methods of communication, power management, and data security had to be implemented for mPOS devices.

## 5 BBPOS TRADE SECRETS

62. Of the broad collection of proprietary information that BBPOS has developed since it began developing mPOS devices in 2008, it includes the following asserted trade secrets detailed in this section.

63. The first three trade secrets relate to BBPOS being able to communicate credit card information between an mPOS device and the mobile phone using the audio jack (i.e. headphone / microphone port) on the Mobile phone. This approach allows BBPOS to communicate digital information (and even encrypted information) by converting the digital information to an audio signal to send to the mobile phone audio jack, and then reconverting that audio signal back to digital information in the mobile phone. These methods include developing both circuitry and software on BBPOS' mPOS devices, as well as developing software on the Mobile phone (within BBPOS' SDK application).

64. Although, at the time, it was unusual to communicate digital information to a mobile phone using an audio jack, BBPOS chose this approach because almost every mobile phone has an audio jack. Almost all mobile phones have the ability to plug in a headphone or a headphone with a microphone for listening to music or making phone calls, respectively. In the 2012 time period, many mobile phones did not support Bluetooth, WIFI, or USB communication methods, which would be the traditional means for communicating digital information. Therefore,

BBPOS chose to use the audio jack method and to ensure their ability to use their mPOS devices with as many mobile phones as possible.

65. Because BBPOS utilized this unusual method of communicating digital information from the mPOS device to a mobile phone, they had to develop several proprietary methods and techniques to ensure a reliable communication method. This required much research, investment, and multiple testing trials to understand all possible mobile phone and payment system formats that could be used with an mPOS device. After spending much time, and financial investment, BBPOS was able to design a single configurable solution to address this market situation. Among the proprietary methods they used to ensure reliable communication through an audio jack, BBPOS developed:

1. **Audio Jack Polarity detection –** determines if the base of the mobile phone's audio jack has a positive or negative polarity and to route the microphone/input signal appropriately. This enables a single solution to support multiple mobile phone signal formats.
2. **Power Management –** methods for efficient power use for battery powered mPOS devices as well as performing sleep and auto wakeup (Power on) functions in order to conserve power.
3. **Signal control settings and auto gain control –** determines the appropriate gain (e.g. signal thresholds) to use in decoding data, and at what speed to reliably transmit and receive the information based parameters defined for the specific mobile phone being used.

66. The next two trade secret categories relate to BBPOS' general experience in POS data encryption and their extensive testing and implementation of the many communication formats used by different mobile payment applications. Principally, these are:

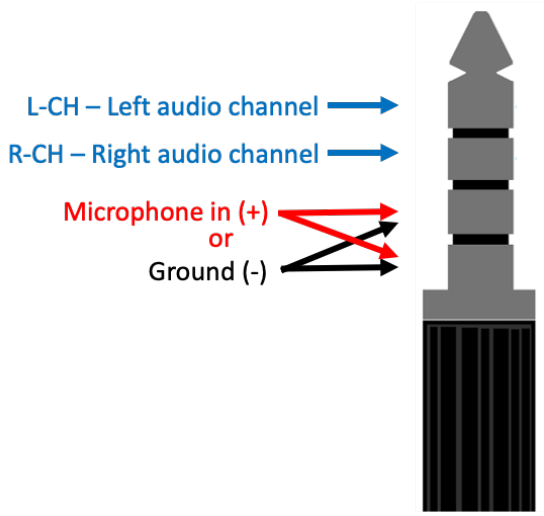
4. **Communication Formats –** over 25 different formats for sending credit card and transaction related information between the mPOS device and the mobile phone to ensure compatibility with different mobile payment vendor applications.

5. **Data Security / Encryption methods –** methods for encrypting credit card data based on variations of data encryption methods.

67. These five categories of trade secrets are described in more detail below.

### 5.1.1 BBPOS' Audio Jack Polarity Detection design

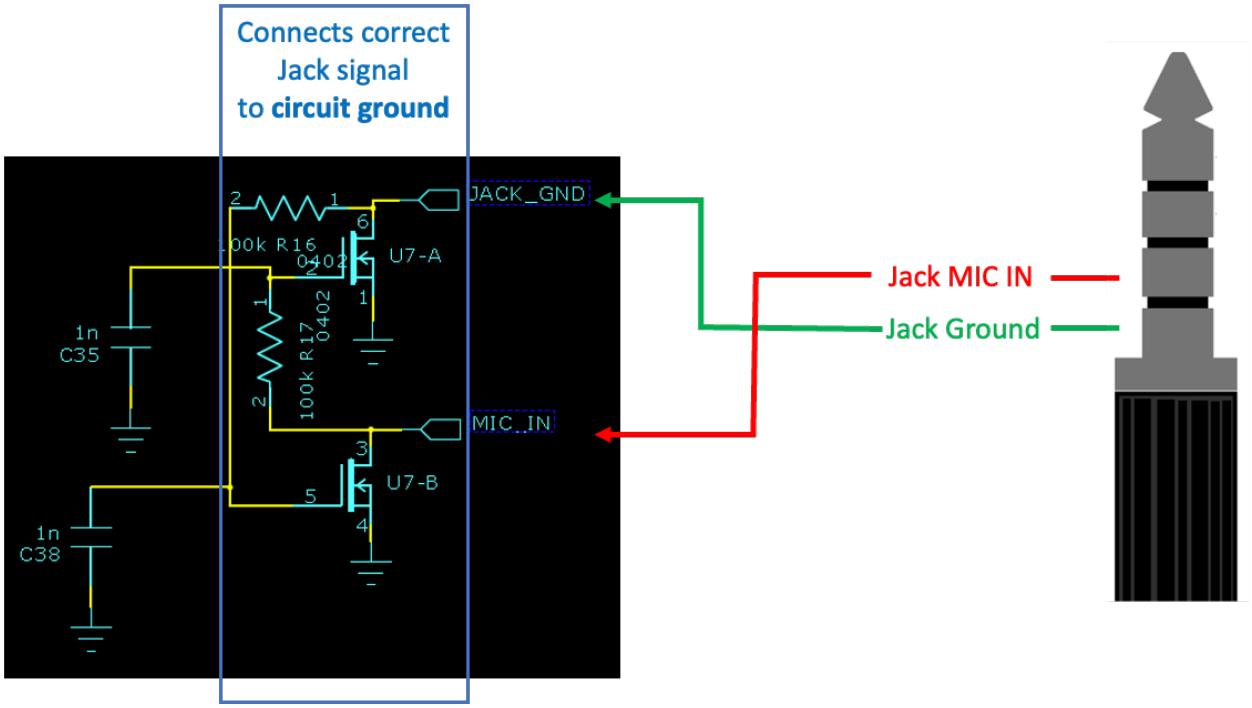
68. The following diagram shows the "rings" of an audio jack that has a left and right audio channel and microphone channel.




***Mobile Phone Audio Jack***

69. BBPOS discovered that one issue communicating through an audio jack is that not all mobile phones (especially in 2012) adhered to the same audio jack standard. BBPOS learned that some phones reverse the polarity of the ground and microphone rings. This can cause communication problems and possible damage to the devices when connecting an active device such as an mPOS to a mobile phone using the audio jack. Therefore, BBPOS developed a method to automatically detect the polarity between the ground and microphone rings and then internally (within the mPOS) reverse those connections to obtain the expected polarity.

70. BBPOS developed several circuit designs to perform this automatic polarity detection and polarity reversal. One of the simplest circuits that performs this detection and reversal is shown below:




***BBPOS Polarity Detection circuit***
**EMVSwiper_PCB1_v0.3.0.pdf pg.5 [BBPOS_1687755-BBPOS_1687762]**

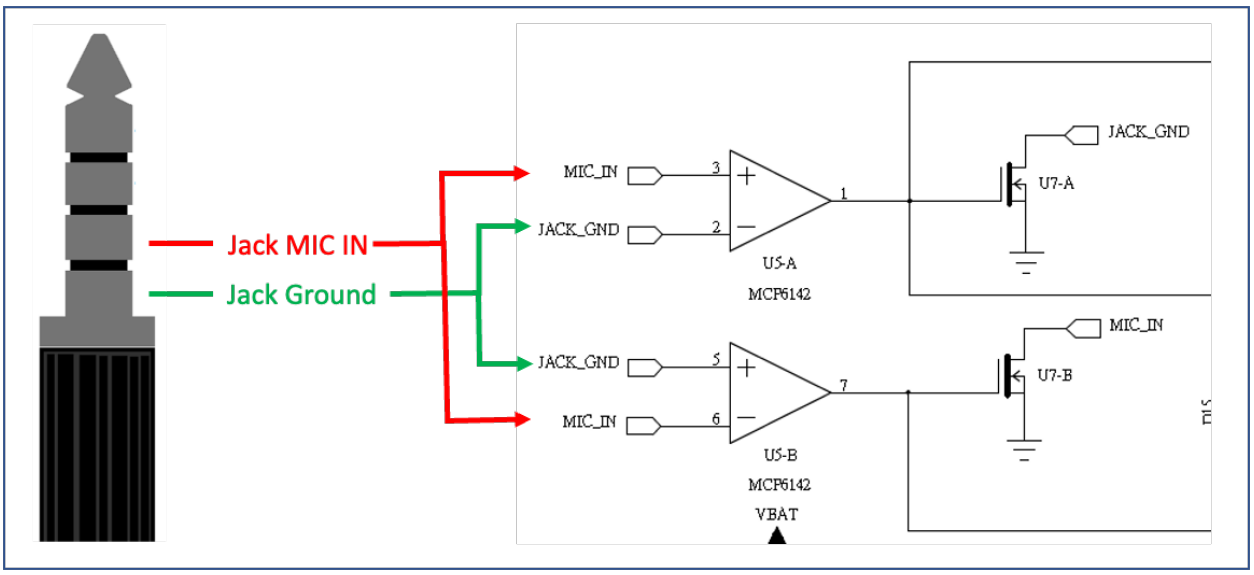
71. In this circuit, if the "Jack MIC IN" coming from the mobile phone has a higher voltage than the "Jack Ground", then the voltage from the Jack MIC IN will turn on (open) the

upper transistor (U7-A). This will connect the Jack Ground to the **circuit ground** of the mPOS device.

72. If the Jack Ground from the Mobile Phone has a higher voltage than the Jack MIC IN, then the voltage from the Jack Ground will turn on (open) the lower transistor (U7-B). This will connect the Jack MIC IN to the **circuit ground** of the mPOS device.

73. Determining whether the Jack MIC IN or Jack Ground has the higher voltage allows the circuit to connect the correct Jack signal to the circuit ground. In doing so, this circuit performs both the Polarity Detection of the Polarity but also the connection of the correct Jack signal to the mPOS' circuit ground. This ensures that the polarity is correct and allows the proper communication of the digitally encoded information between the mobile phone and the mPOS device as well as preventing any damage to either device.

74. BBPOS developed other versions of this circuit to detect and reverse the polarity of the ground and microphone channels. The other BBPOS designs follow the same design concept as the previous circuits above, simply using different components or configurations. For example, the following is another BBPOS variation of a circuit to detect and reverse the polarity of the ground and microphone input. This variation uses a pair of N channel MOSFET transistors to connect the correct Jack signal to **circuit ground** as does the first circuit above. This variation however uses a pair of OpAmps (U5A & U5B on the left) to first detect the polarity of the Jack MIC IN and the Jack Ground.




***BBPOS Polarity detection and reversal circuit***

***Paypal-PCB1-ST04-V3.1.pdf [IngenicoInc_0009727-IngenicoInc_0009729]***

75. BBPOS incorporated their Polarity Detection circuit design in several of their mPOS devices including the GX series (e.g. G4X, G5X).

### 5.1.2 BBPOS' Power Management design (Auto Power On)

76. Various methods have been developed to optimize the power usage of many types of battery powered electrical devices. However, since very few companies in 2012 had developed mPOS devices that communicated via the mobile phone's audio jack, very few, if any, power management methods had been developed for mPOS devices, especially those using audio jack interfaces, except for BBPOS.

77. BBPOS has designed many efficient mPOS functional circuits as well as efficient power supply circuits for their devices. One of BBPOS' power management designs allows the microprocessor to put the mPOS device to sleep whenever the device is not active and then to "wake up" or automatically turn on the device when needed.

78. There can be several methods for determining when to automatically turn on or wake-up the device. However, the method must determine the correct trigger or "trigger threshold" to use that always reliably wakes up the device when a card transaction will be performed.

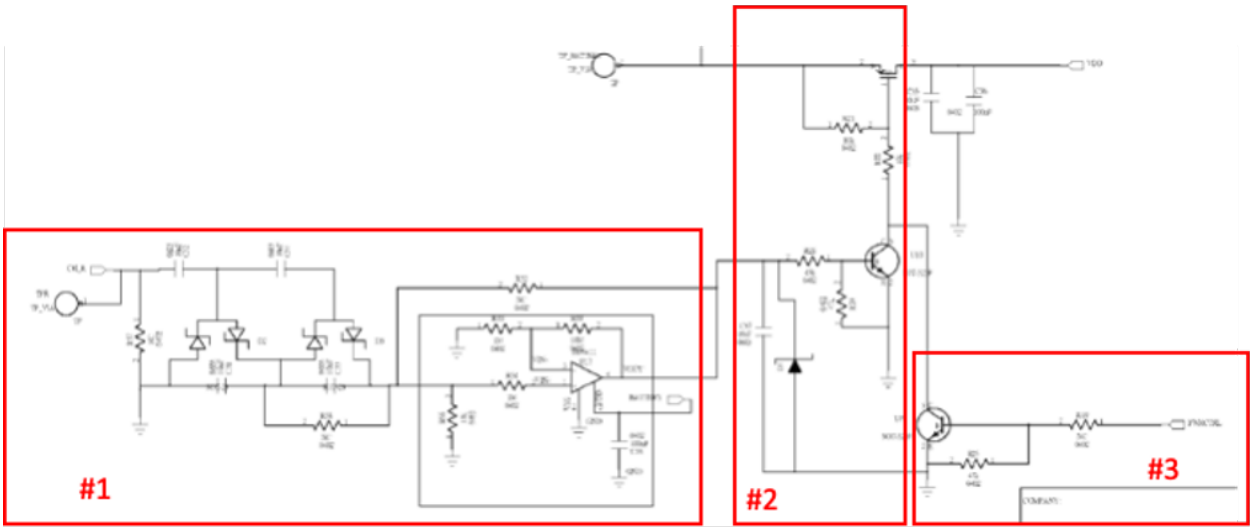
79. As part of their power management solution, BBPOS designed a "temporary trigger" that determines when the mobile phone is plugged into the mPOS device and when there is any activity on the audio jack interface (e.g. on the right or left audio channel). During this temporary wake-up period the microprocessor allows the power to stay on, at least for a short period, using a "permanent trigger". During this temporary wake-up period the microprocessor determines if there is activity on the audio jack to indicate an mPOS initialization or card transaction from the mobile phone. If there is no "valid" activity on the audio jack interface (i.e. no mPOS initialization or transaction), then the mPOS device will power down by turning off the "permanent trigger".

80. For example, if you were to play music on the mobile phone through the audio jack interface to the mPOS device, the "temporary trigger" would turn on the power to the microprocessor, however as soon as the microprocessor determined that there was no valid mPOS signal, it would immediately turn off the power and put the mPOS device back to sleep.

81. Therefore, BBPOS' design will automatically power-on or "wake-up" the mPOS whenever the mobile phone sends a valid initialization or transaction to the mPOS device. In summary, BBPOS' power management for their mPOS device includes up to three different types of circuits as follows:

1. **the "temporary trigger" circuit** - to temporarily wake-up the microprocessor when the mobile phone audio jack is plugged into the mPOS device and when there is activity on the audio jack interface.
2. **the "power switch" circuit** – this circuit enables battery power to microprocessor and the card reader and can also be controlled by the microprocessor using the permanent trigger.
3. **the "permanent trigger" circuit** – when the microprocessor is powered up by the power switch (e.g. after being activated by the temporary trigger), the microprocessor enables the "permanent trigger" to keep the power on while determining if there is a valid transaction or initialization being requested from the mobile phone on the audio jack interface. If there is no "valid" activity on the audio jack interface the mPOS device will go back to sleep when the microprocessor disables the permanent trigger. If there is valid activity on the audio jack interface, the microprocessor keeps the permanent trigger enabled for the duration of the transaction.

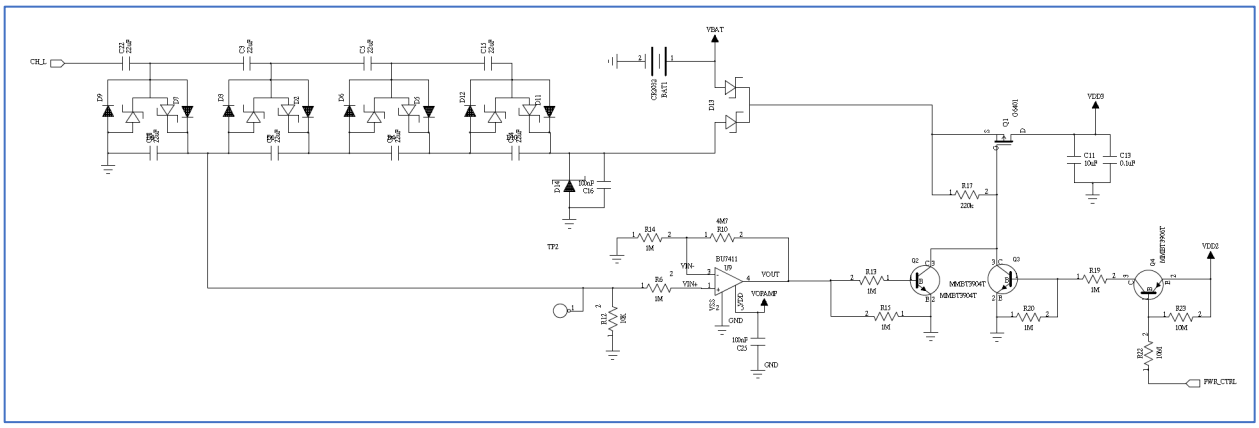
82. BBPOS experimented with various versions of each of these circuits that would eventually work reliably with the many different types of mobile phones. The following diagram illustrates one of the initial designs of these three circuits.




*EMV_SWIPER.pdf pg. 5 [IngenicoInc_0135063-IngenicoInc_0135068]*

83. In this diagram, Circuit #1 (the "temporary trigger") determines if the mobile phone is plugged into the audio jack and if there is any type of activity on the audio jack through the R-CH (right audio channel). Circuit #2 (the "switch") powers on (wakes up) the mPOS device either when the temporary trigger or permanent trigger is enabled. Circuit #3 (the "permanent trigger") allows the microprocessor to keep the power to the mPOS on or turn it off.

84. As mentioned above, BBPOS had developed several variations of this circuit with different characteristics. For example, the following is another variation from BBPOS that contains the same three circuits as above but also allows the power contained in the audio signals to augment the battery of the mPOS.



***PayPal-PCB1-ST04-V3.1.pdf Pg. 3 [IngenicoInc_0009727-IngenicoInc_0009729]***

85. Other BBPOS design examples include Paypal-PCB1-ST04-V1.0.pdf [BBPOS_1687763-BBPOS_1687765] which provides more audio boost & a faster turn on, and Swiper-PCB1-ST11-v2.0.pdf [BBPOS_1687766-BBPOS_1687768] which provides more audio boost but using less power. In all cases the design concept is the same, which is to wake-up the mPOS device when there is activity on the audio jack interface (on the R or L audio channels), and to keep the power on if there are valid mPOS signals on the audio jack interface.

### 5.1.3 BBPOS' Pre-analyzed communication settings and adaptive threshold (or Auto Gain Control)

86. BBPOS has tested many different mobile phones and determined that the power and signal performance of the various phone's audio jacks differ widely from one phone model to the next. One reason for this is that mobile phone manufactures generally do not design their audio jacks to have the precision necessary to transmit and receive audio-encoded digital information. They are designed only to transmit and receive music and voice audio signals.

87. As such, BBPOS has had to test and analyze the performance of the audio output and input of many different mobile phones. From that, BBPOS has developed both a set of

communication parameters for different types of phones as well as an algorithm to compensate for the lack of precision in the mobile phones' audio characteristics.

88. The set of communication parameters developed by BBPOS were created by testing and analyzing the performance of each phone model using different settings. Then an optimum set of parameters was selected for each phone model. I will refer to these parameters as BBPOS' "***pre-analyzed audio and communication settings***".

89. The algorithm that BBPOS developed to compensate for the variety of audio characteristics of the different phones requires a real-time analysis of the audio signals and a method to adaptively detect the signal transitions within the audio signal. I will refer to this algorithm as the "***Adaptive threshold method"***. BBPOS refers to this feature in their design documents as Automatic Gain Control (AGC) because it determines or predicts the amplitude (gain) of the incoming audio signals.

90. Details of each of these two aspects of BBPOS' communication trade secrets are below.

#### *5.1.3.1 BBPOS' pre-analyzed audio and communication settings*

91. For the pre-analyzed device settings, BBPOS has tested and determined 11 different parameters that need to be configured for the audio and communication settings for more than 442 different phone models.

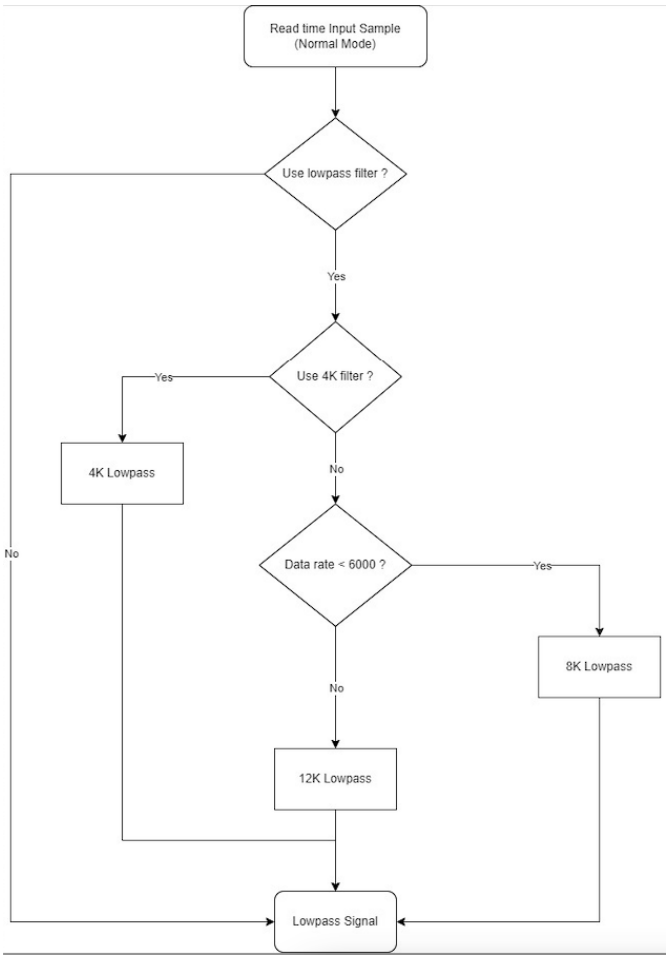
92. The following are the different parameters to be configured for each phone model:

- Max Output Frequency – determines the maximum frequency or speed to transmit data for each phone model
- Max Input Frequency - determines maximum frequency or speed to receive data for each phone model

- Output Volume Offset – the relative volume offset for each phone model
- Output Volume Adjust Delay – determines if a particular phone model requires a delay in volume adjustment
- Preamble Length – determines the length of the preamble symbols used at the start of a transmission
- Postamble Length - determines the length of the postamble symbols used at the end of a transmission
- Low Volume – identifies the recommended minimum volume setting
- Decoder Mode – determines the type of data Decoder Mode
- Decoder Option - determines the data Decoder Options
- Microphone Audio Source – audio source of the microphone
- Output 11k Mode – Determines if the device can use a high speed data transmission

#### *5.1.3.2 BBPOS' adaptive threshold method* (Automatic Gain Control)

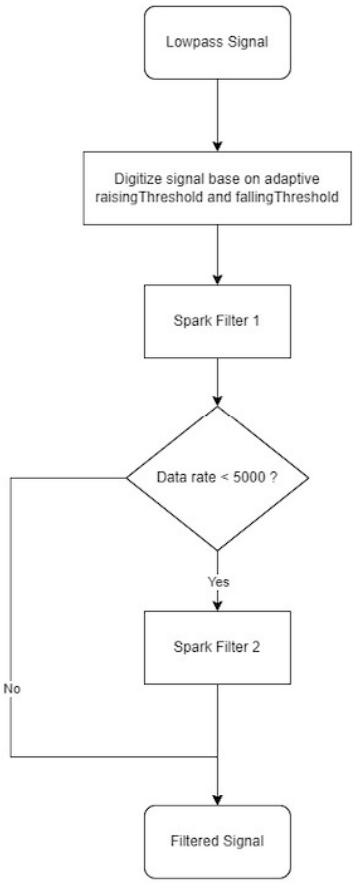
93. BBPOS' adaptive threshold method occurs as part of the overall processing and conditioning of the incoming audio signal. The flow of this process is described in 3 parts as shown in the 3 flowcharts below.




***Adaptive threshold flowchart – pt.1***

***Adaptive threshold flowchart.pdf [BBPOS_1687849]***

94. In Part 1 above, the incoming audio signal is first sent through a lowpass filter based on the maximum rate at which the mobile phone can communicate. The produces a "Lowpass Signal" which is sent to Part 2.




***Adaptive threshold flowchart – pt.2***

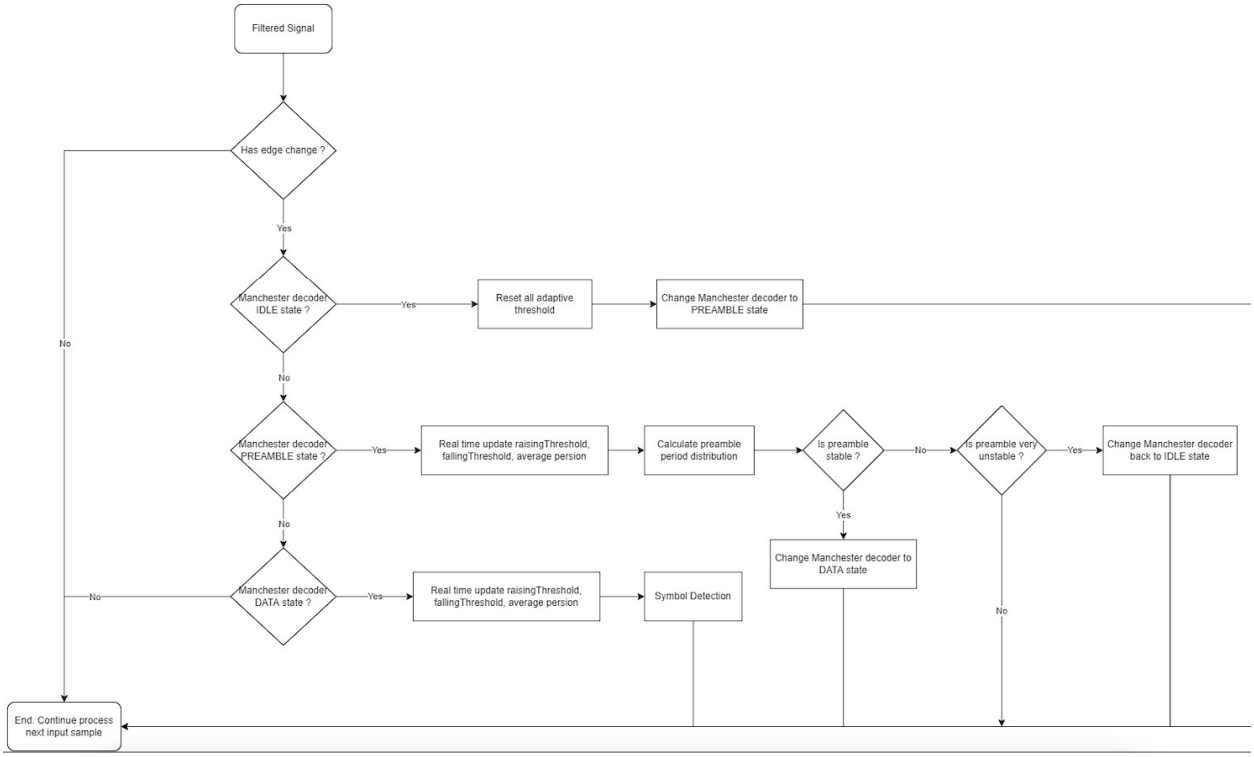
***Adaptive threshold flowchart.pdf [BBPOS_1687849]***

95. In Part 2 above, the filtered Lowpass signal is then processed using the "adaptive threshold method". This is depicted in the flowchart above in the "Digitize signal base on adaptive raisingThreshold and fallingThreshold". In this method the system identifies the rising and falling points (i.e. reversals) of the audio signal and calculates a threshold for the local minimum and local maximum. This is a moving average based on a certain number of audio wave transitions or symbols within a running window. This allows the system to better predict where the next wave transition (that contains valid symbol info) will start.

96. This is necessary because, as mentioned above, different phone models have different performances characteristics of their audio input and output. For example, different

phones have different signal amplitudes where the DC offset can also slowly change or drift over time.

97. Once the adaptive threshold method is performed there is an additional spark filter applied depending on the communication speed, which filters out narrow signal glitches which can occur from some phones. From here the "Filtered Signal" is sent to Part 3.




***Adaptive threshold flowchart – pt.3***

***Adaptive threshold flowchart.pdf [BBPOS_1687849]***

98. In Part 3 above, the incoming audio signal has been processed to the point that the software can begin extracting data (i.e. decoded the signal). For example, from the signal that has been processed in Part 1 and Part 2 above, it is much easier and reliable to determine the start (edges) of the audio signal that actually contains data as opposed to noise or losing the signal

transition due to amplitude drift. From there, the software determines if the audio signal contains a Preamble (a test signal indicating the beginning of transaction data), the actual transaction data, or whether it is in an idle state. As each symbol in the audio signal is being detected (either the data symbols or the preamble symbols), the adaptive threshold method adjusts the moving average of the signal transitions to compensate for the drift of the audio signal.

99. The parameters of the adaptive threshold method are reset whenever the audio signal is idle, to prepare to restart the adaptive calculations for the next set of audio signals.

### 5.1.4 BBPOS' proprietary mPOS communication formats

100. Mobile payment applications are used by vendors that run on a mobile phone and interface with the mPOS devices to retrieve payment / card information. A popular mobile payment application for example is Square.

101. BBPOS had developed extensive experience with many Mobile Phone Payment apps from many different companies that they could interface their mPOS devices with those mobile payment applications. As such, BBPOS created a collection of different communication formats specifying exactly what data fields need to be communicated and in what format. By the beginning of 2013 BBPOS had developed over 25 different communication formats compatible with different companies and mobile applications.

102. For example, BBPOS' Format_ID_7 and Format_ID_21 are communication formats that are compatible with the PayPal mobile payment application.

103. BBPOS has documented these communication formats for licensed SDK users (i.e. developers that plan to interface with BBPOS devices with mobile phone applications). The document is called BBPOS Data Output Format for POS device. This document continues to grow as BBPOS collects new formats. As of May 3, 2012 (version 1.15 of the document) BBPOS

defined specifications for 9 different communication formats. As of August 27, 2013 (version 1.40) BBPOS had defined a total of 25 communication formats.

104. In addition to the communication formats that identify the specific data fields that must be communicated and in what format, BBPOS has also defined the patterns of data within the card data fields. Although data within the credit / debit card data fields are standardized, BBPOS has the ability to communicate standard credit card data as "masked" or "tokenized". In addition, BBPOS has the ability to support non-standard cards such as gift cards or ID cards.

105. The following provides an example of one of the communication formats defined by BBPOS:

**4.1.6. Format 7**

| 1 byte | 1 byte | 2 byte | 2 byte | 2 byte | 26 byte | 10 byte | 40 byte | 1 byte |
|---|---|---|---|---|---|---|---|---|
| Format ID | Length of PAN | First 4 digits of PAN | Last 4 digits of PAN | Expiry Date | 26 bytes after the 1st ^ | KSN | Encrypted Track 1 | Checksum |
| (0x07) | | | | (YYMM) | | | | CRC8 of all bytes |

- Each character in Track 1 is 6 bits in length. 4 characters are packed into 3 bytes and padded with zero to make it 40 bytes in length before encryption.
- 26 bytes after the 1st ^ are extracted from Track 1 before encryption.
- CRC is using CCITT CRC.
- This format is for Swiper with magnetic head to read track1 only.

***BBPOS-DataOutputFormat-V1.21.pdf pg. 8 [BBPOS_0005649-BBPOS_0005663]***

106. The following is a portion of the BBPOS code that determines what communication format is being used (12 in this example) and the various card data patterns associated with that.

```
if (formatID == DataOutputFormat.FORMAT_ID_12.getID()) {
      // JIS II T card
      int tcardPattern = getTcardPattern(ucResult);
      decodeData.put("tcardPattern", Integer.toString(tcardPattern));

      if (tcardPattern == 1 || tcardPattern == 3) {
           decodeData.put("encTrack",    getEncTrack(ucResult));
```

```
        decodeData.put("maskedPAN",   getMaskedPAN(ucResult));
        decodeData.put("expiryDate",  getExpiryDate(ucResult));
       decodeData.put("tcardMembershipID",getTCardMembershipID(ucResult)
);
       //decodeData.put("cardholderName",  getCardholderName(ucResult));
       decodeData.put("partialTrack",      getPartialTrack(ucResult));
       decodeData.put("ksn",               getKSN(ucResult));
                    }
else if (tcardPattern == 4) {
       decodeData.put("encTrack",    getEncTrack(ucResult));
       //decodeData.put("maskedPAN", getMaskedPAN(ucResult));
       //decodeData.put("expiryDate", getExpiryDate(ucResult));
       decodeData.put("tcardMembershipID",getTCardMembershipID(ucResult)
);
       //decodeData.put("cardholderName",  getCardholderName(ucResult));
       decodeData.put("partialTrack",      getPartialTrack(ucResult));
       decodeData.put("ksn",               getKSN(ucResult));
                    }
else if (tcardPattern == 5) {
       decodeData.put("encTrack",    getEncTrack(ucResult));
       //decodeData.put("maskedPAN", getMaskedPAN(ucResult));
       decodeData.put("expiryDate",  getExpiryDate(ucResult));
       decodeData.put("tcardMembershipID",getTCardMembershipID(ucResult)
);
       decodeData.put("cardholderName",    getCardholderName(ucResult));
       decodeData.put("partialTrack",      getPartialTrack(ucResult));
       decodeData.put("ksn",               getKSN(ucResult));
                    }
else if (tcardPattern == 6) {
       decodeData.put("encTrack",    getEncTrack(ucResult));
       //decodeData.put("maskedPAN", getMaskedPAN(ucResult));
       //decodeData.put("expiryDate", getExpiryDate(ucResult));
       decodeData.put("tcardMembershipID",getTCardMembershipID(ucResult)
);
       //decodeData.put("cardholderName",  getCardholderName(ucResult));
       decodeData.put("partialTrack",      getPartialTrack(ucResult));
       decodeData.put("ksn",               getKSN(ucResult));
}
```

***SwiperDecoder.java [BBPOS_0691264-BBPOS_0691272]***

### 5.1.5 **BBPOS' Data Security / Encryption Methods** (DUKPT Data method)

107. The payment industry has many standards and methods for providing data security and encrypting data. There are common encryption methods specifically used for retrieving and transmitting credit card information. For example, TDES (Triple Data Encryption Standard) is used for protecting and retrieving card information from an EMV credit card (cards that contain

the smartchips). Another well known method is the procedure for producing or deriving the keys for encrypting and decrypting information, also known as "key injection". For example, DUKPT (Derived Unique Key Per Transaction) is used to derive the keys for encrypting or decrypting a user's PIN code, where the key only has a one-time use.

108. As just mentioned, TDES is the typical method for encrypting card transaction data, where DUKPT is the method for deriving keys for PIN encryption. However, BBPOS has developed a proprietary method, that is a modification of the standard DUKPT methods, for encrypting card transaction data. Specifically, BBPOS' method allowed for the derivation of a data encryption key using their own DUKPT method for deriving the encryption/decryption key, so that data could be encrypted locally on the mPOS device to be sent to the server to be decrypted in a secure manner.

109. The following shows the ANSI standard for deriving keys for PIN encryption. The standard methods for deriving keys for PIN encryption involve applying the PIN encryption variants, shown below, to the Base Derivation Key (BDK).

The key variant constant of TDES-Dukpt is as followed:

| Key used for | Variant constant | |
|---|---|---|
| | Variant constant-L | Variant constant-R |
| PIN Encryption | 00 00 00 00 00 00 00 FF | 00 00 00 00 00 00 00 FF |
| Message Authentication, request or both ways | 00 00 00 00 00 00 FF 00 | 00 00 00 00 00 00 FF 00 |
| Message Authentication, response | 00 00 00 00 FF 00 00 00 | 00 00 00 00 FF 00 00 00 |
| Data Encryption, request or both ways | 00 00 00 00 00 FF 00 00 | 00 00 00 00 00 FF 00 00 |
| Data Encryption, response | 00 00 00 FF 00 00 00 00 | 00 00 00 FF 00 00 00 00 |

Table 1 Variant constants for transaction keys

***ANSI+X9.24-1-2009.pdf pg.56***

110. The above table also shows the ANSI standard for deriving keys for Data encryption. The standard methods for deriving keys for Data encryption involve applying the

Data encryption variants, above, to the Base Derivation Key. Then the result of applying the Data variants is then self-encrypted using the TDES method. This process is shown below.

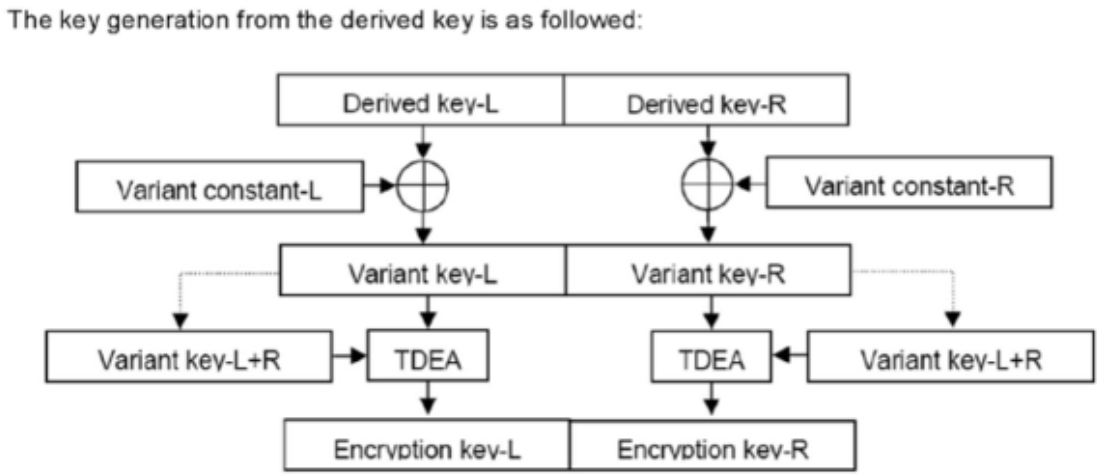



Figure 14 Key calculation for Data Encryption keys

*ANSI+X9.24-1-2009.pdf pg.56*

111. BBPOS has created two different techniques for encrypting data that diverge from the ANSI standards described above, which they refer to as their DUKPT data encryption methods. Specifically, these are as follows:

a. Deriving a key for encrypting data that uses the standard method for deriving the PIN key. In other words, rather than use the standard method for deriving a key for encrypting data using the data variants, BBPOS derives a key for encrypting data **using the PIN variants**.

b. Deriving a key for encrypting data that uses the standard variant for the data key **without performing the TDES self-encryption of the result of the applied Data variants**. This provides a more efficient means of establishing an encryption key with the same or similar security as does the PIN key derivation.

112. BBPOS also incorporated their proprietary data encryption methods into communication formats they defined, as described in the previous section. For example, in BBPOS' specification of communications format #11, shown below, it indicates that the BBPOS' method of "DUKPT data key" derivation (i.e. method b. above) will be used for the data encryption.

#### 4.1.9. Format 11

| 1 byte | 1 byte | 2 byte | 2 byte | 2 byte | 26 byte | 10 byte | 1 byte | 80 byte | 1 byte |
|---|---|---|---|---|---|---|---|---|---|
| Format ID (0x0B) | Length of PAN | First 4 digits of PAN | Last 4 digits of PAN | Expiry Date (YYMM) | 26 bytes after the 1st ^ | KSN | Track 1 Length | Encrypted Track 1 | Checksum CRC8 of all bytes |
| 0 | 1 | 2 | 4 | 6 | 8 | 34 | 44 | 45 | 125 |

- Track 1 is in ASCII.
- Track 1 is padded with 0x00 to the least multiple of 8 bytes before encryption.
- Encrypted track is padded with 0x00 to 80 bytes.
- DUKPT data key is used for encryption.
- TDES CBC mode is used to encrypt track 1.
- CRC is using CCITT CRC.
- This format is for Vantiv with magnetic head to read track1 only.

***BBPOS-DataOutputFormat-V1.40.pdf***

## 5.2 BBPOS' protection of the asserted trade secrets

113. BBPOS contracts included language to maintain confidential information as protected and that rights to the products were non-transferrable and non-assignable even after sale of the contracting parties to another entity. All terms of the contracts would remain in effect for 5 years after termination of any such agreement. Refer to Section 3.2 for the Agreement Language.

114. In addition, BBPOS documents were created with the designation of "BBPOS Confidential and Proprietary" such that all parties receiving them would be made aware that the information contained therein should be protected as such.

115. All clients of BBPOS are required to sign a NDA to maintain confidential information as confidential, such as the contents, codes, formats, APIs and SDK. As described by BBPOS in deposition:

> *Q. Okay. Let me -- I'm not sure you understood the question, because it wasn't a yes or no, so let me try again.*
>
> *What steps are taken by BBPOS to ensure that when they give the Swiper API programming guide to their customers that the Swiper API programming guide will remain confidential for BBPOS?*
>
> *A. We will have them to sign NDA before we release the API programming guide to the customer.*
>
> ***Ben Lo's Deposition, Dec 08, 2021, Pg. 105***

116. In addition, BBPOS protected their documentation of trade secrets by the following methods:

- All BBPOS documents, trade secret or otherwise, are notated with the designation of "Confidential and Proprietary BBPOS Limited" in the headers and footers of each page.
- All trade secret documentation and physical devices are kept in an area accessible only to those employees with a need to know.
- Employees are required to sign an Employee Agreement stating they understand and will comply with confidentiality to protect the trade secrets of BBPOS.
- BBPOS shared trade secret documentation with clients only via email; they did not provide online access to any trade secret documentation.
- Any physical devices sent to laboratory facilities for testing and certification purposes were protected by signed NDA with laboratory and shipped via courier.

117. Therefore, BBPOS took appropriate steps to secure its assets and trade secrets including through contract terms, Non-Disclosure Agreements, Employee Agreements, physical control of devices and document stamping.

# 6 BBPOS SHARED TRADE SECRETS WITH ROAM/INGENICO

## 6.1 Information Sharing Timeline and Relationship Activities

118. Over the course of 2012, the relationship between BBPOS, ROAM Data and Ingenico evolve as the race delivering an mPOS device heats up. BBPOS has a long relationship with ROAM and enters into an Engineering and License Agreement with ROAM to build and deliver an mPOS device specifically for ROAM. Ingenico has an investment in ROAM; Ingenico also owns Landi, a competitor in the POS device market. ROAM looks to acquire ROAM but must have Ingenico approval. BBPOS is open to acquisition and supports activities associated with due diligence. Meanwhile, BBPOS continues development of mPOS capabilities per the agreement with ROAM. A summary of activities and information sharing follows:

## 6.2 BBPOS trade secrets shared with ROAM

119. The agreement between ROAM and BBPOS was for the development and production of an mPOS device to marketed as the ROAMpay device starting in 2010. This would be based on the BBPOS G3X version with enhancements to take advantage of newer technology and to support an expanded set of cell phone models. As of Feb'12, Christopher Rotsaert was brought in as the new Product Manager for the ROAM/Ingenico mPOS solution. Rotsaert began his assignment by meeting the BBPOS team, ROAM's partner for producing their POS devices. To get started with the introduction and working relationship, Rotsaert requested information vis email on the mPOS development and roadmap, specifically:

- Legacy portfolio of BBPOS in term of products, reference design, security (physical, logical), applications
- Development activities by BBPOS for Roam Data in scope of readers (hardware, firmware), mobile application, gateway /servers, keys management
- R&D developments roadmap: hardware, firmware & security roadmap for swipe / swipe+Cless[4] & swipe+chip+Cless readers
- Security updates between current swipe reader & next swipe+Cless & swipe+chip+Cless readers
- Topics with possible support of Ingenico to fasten development

***Email from Rotsaert to BBPOS Team [BBPOS_0646802-BBPOS_0646804 emphasis added]***

120. Through August of 2012, BBPOS transmitted proprietary information in the form of schematics, design documents, source code, email descriptions, etc. on many occasions as part of this joint project. In addition to information shared in requirements, specifications, designs and schematics, countless emails were exchanged between the BBPOS, ROAM and Ingenico development teams to help with the design and development of device APIs, audio jack circuits and characteristics, power management of the device, Communication formats, and DUKPT data encryption methods. BBPOS also provided software SDKs containing mPOS interface functionality on a continuing basis throughout their engagement with ROAM Data as versions were updated with new formats, phone settings and APIs for new features.

[4] Cless stands for Contactless as for a "tap and pay" or NFC (Near Field Communication) card

121. In Feb'12, Jimmy Tang of BBPOS sent an email attaching DUKPT data encryption source code and Swiper Communication / Output Format Specifications to Christopher Rotsaert, of Ingenico BBPOS_0004382. This is comprised of application code, include files and libraries along with design docs. The attachments were:

- DukptClientTest.zip (DUKPT Data methods)
- ROAM Swiper Output Format 10.doc (Communication Formats)

122. As a response to the email, Jerome Grendemenge, Ingenico, asks for and receives assistance from, Jimmy Tang, BBPOS, on how to use the Swiper Controller APIs to help troubleshoot Ingenico's development of an interface with the BBPOS swiper using the DUKPT data encryption and Communication Formats [BBPOS_0004422-BBPOS_0004423].

123. Late in Mar'12, ROAM Data CEO introduces, John Chiu as the new Engineering manager to work with BBPOS. As such Ben Lo and the BBPOS team begin working with John Chiu to deliver the products for prospective clients including PayPal. Ben Lo, BBPOS, shares Communication Format information with John Chiu and the ROAM engineering team [BBPOS_163221] to help with their testing of the APIs they are building. Ben Lo, BBPOS shares both the specification and sample inputs for data Communication format IDs 7, 10, 17 & 20, DataFormat2Server-V1.1.docx [BBPOS_1632220-BBPOS_1632225], as well as a simulation program, SwiperSimulator.exe [BBPOS_1632240] to assist them in their testing.

124. Over the course of 2012, BBPOS attended numerous meetings, workshops, presentations, etc. that shared BBPOS' proprietary information all under the alleged purpose of speeding up the development of mPOS by leveraging the Ingenico resources, even though Ingenico had a more limited experience with designing mPOS devices than BBPOS. Ingenico was an investor in ROAM Data at the time and was also party to the acquisition activities that

ROAM had initiated with BBPOS. ROAM had a longstanding relationship with BBPOS for their Swiper mPOS, specifically the G3X device which uses the audio jack interface of a mobile phone. Ingenico had an R&D group in Valence, FR for their broad portfolio of POS terminal devices, most of which were hard-wired machines and none of which were audio interface enabled. Ingenico also owned Landi[5], a POS development shop who also had a portfolio of POS products, again mostly wired and none using the audio jack interface of a mobile phone.

125. While BBPOS had focused on the Swiper mPOS development since 2008, they had also begun development and testing of their EMV (chip card) devices.

126. Several ROAM/BBPOS demos were performed during this timeframe for prospective clients such as PayPal, NAB, Google, PayPass, and Chase among others. In late March the Cartes Asia expedition was held and the mPOS joint PM Christopher Rotsaert reached out to Ben Lo from BBPOS to meet at the tradeshow. He indicated Ingenico's wholly owned subsidiary, Landi, would have technical people at the show and he wanted "to discuss the opportunity to use Landi low-cost platform supporting embedded EMV L2 for chip & Cless." [BBPOS_1682180-BBPOS_1682184]

127. In this same email conversation, Rotsaert indicated that the platform being discussed was "quite new in Landi and not all features are yet available: in particular, swipe, audio jack interface & power management are not ready. I'd like to discuss with you at Cartes & next week on how we could manage to move fast leveraging both teams. Liu Shying, R&D VP of Landi will be at Cartes on Wednesday & Thursday, I'd like to organize a discussion together on this opportunity." [BBPOS_1682180-BBPOS_1682184, emphasis added]

[5] Ingenico acquired 100% shareholding of Landicorp in 2012 [https://www.businesswire.com/news/home/20190613005239/en/China-Financial-POS-Terminal-Industry-Report-2019-2025]

128. BBPOS began providing information to the ROAM Data team in Feb 2012 to continue development on the next version of EMV and NFC enhancements as well as the PayPal version. In addition, BBPOS was very active in other prospective client interactions with ROAM in this time frame, which include Google, NAB, and others such as creditcall, eventbrite, Flexigroup/Paymate, Paypass, Chase paymentech, TMobile.

129. Items being sent from BBPOS to the ROAM Data team and/or directly to Christopher Rotsaert of Ingenico included:

- Multiple Versions of Communication Formats [BBPOS_0649335-BBPOS_0649337, BBPOS_0005646, BBPOS_0005630, and BBPOS_0005121-BBPOS_0005122]
- Explanation of data rates for the audio jack interface and Automatic Gain Control [BBPOS_1396262-BBPOS_1396263 and IngenicoInc_0009756-IngenicoInc_0009757]
- Audio Interface Design Schematics [BBPOS_0005630]
- Multiple versions of ROAMpay API Technical Specs [BBPOS_0004723-BBPOS_0004724 and BBPOS_0004850-BBPOS_0004854]
- DUKPT data encryption/decryption algorithms [BBPOS_0004382]
- Battery & Power management design [BBPOS_0005646]
- PayPal mPOS schematics [BBPOS_0005664]

130. In Apr'12 Ben responds to request from Rotsaert with a discussion regarding the data rate for the audio jack communication with various smartphones - additional emails between Daniel Tsai and Ben (BBPOS) answer in detail Rotsaert's questions that Ben then forwards.

[BBPOS_1396262-BBPOS_1396263] These discussions include regarding power management, and high data rates for the audio jack interface requiring algorithms for Automatic Gain Control.

131. By May'12, ROAM data and PayPal sign the agreement to produce the first PayPal mPOS based on the ROAM Data (aka BBPOS platform). BBPOS starts with the G4X platform already in production and creates a new version for testing with PayPal. Development and testing with PayPal continues and a new form factor is selected for PayPal. This new product evolution becomes the G5X using the PayPal triangle form factor.

132. At this time, Rotsaert decides he will take over aspects of the development of the G5X and convinces the ROAM CEO to direct BBPOS to include him on all BBPOS activities and communication. BBPOS follows through with providing information and engineers to assist as they have been told that Rotsaert is taking over Product Management including the Physical side of the product and to include him on all activity and communication. Coupled with the assumption that an Acquisition of BBPOS was in process, the BBPOS team complies with all requests to provide information on their product designs and status.

133. In May'12, Jimmy Tang, BBPOS, shares information on the new ROAM pay API, RoamPay API 4.0 v1.2 - Client Side Technical Specs.docx, with the ROAM and Inegnico teams and explains to recipients how he is collecting data, e.g. in what type of parameter <String, String> and then gives an example code snippet in Java "hashMap.put(ApiParams.Amount, "1.00"); // amount field" [BBPOS_0004723-BBPOS_0004724]

134. As May'12 comes to a close, Rotsaert sets up a workshop at Valence (FR) and requests BBPOS send their firmware engineer(s) to attend. In addition, Rotsaert requests "all materials you have about swipe + EMV Chip product. Schematic, Your selected core IC specification, Software synoptic... Battery capacity... intermediate milestones …"

[BBPOS_0000003-BBPOS_0000005] The BBPOS teams continues to answer questions regarding antenna size, battery requirements, contactless capabilities, chipsets and other power management topics. [BBPOS_0005186-BBPOS_0005188, IngenicoInc_0010296-IngenicoInc_0010306 and IngenicoInc_0009879-IngenicoInc_0009882].

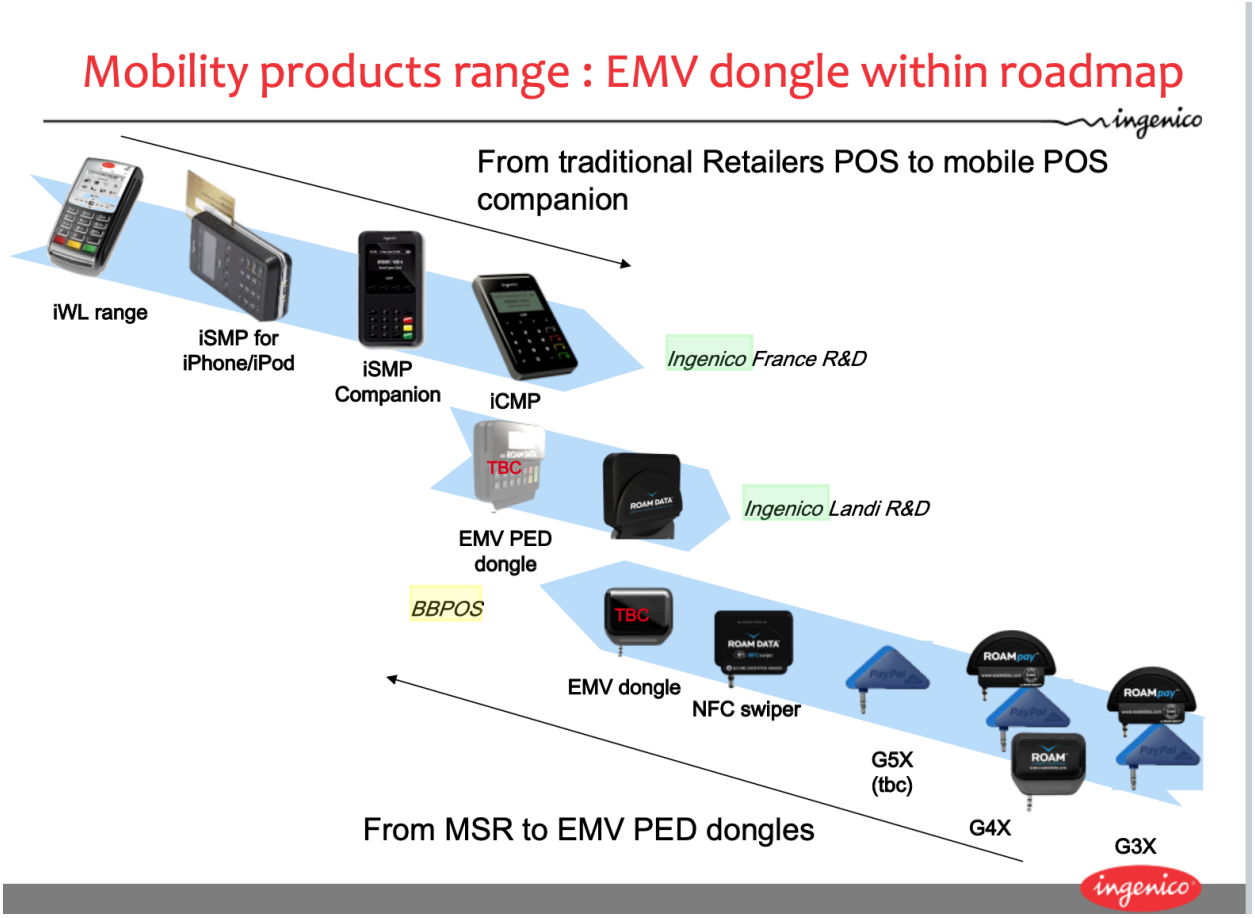
135. As part of the discussion on current status and preparing a joint workshop Rotsaert indicates to BBPOS that his plan is to "release this device before end of the year with a Press Release during Cartes Paris (6-8 November 2012)." [BBPOS_0000003-BBPOS_0000005] Rotsaert then directs BBPOS to participate in developing 2 scenarios – One on Ingenico platform and one on BBPOS platform. He provides direction on what he needs from BBPOS to do so. [IngenicoInc_0010311-IngenicoInc_0010312]

136. In July'12, Rotsaert requests more copies of documentation, schematics and designs from the BBPOS team. And the BBPOS team responds throughout July'12 with all the previously provided designs, specs, schematics and answers to questions regarding same. [BBPOS_0005112, BBPOS_0005646, IngenicoInc_0010655-IngenicoInc_0010656, and BBPOS_0005664 and BBPOS_0005630] One such request from Rotsaert comes as part of an effort to propose a solution for an EasyCash (ING Germany) demo, to which BBPOS replies with their latest version of their Swiper Data Output Format. [BBPOS_0005121-BBPOS_0005122]

137. In Oct'2012, Rotsaert presents a SOW to Landi for development of the iTMP product line which includes power management, security and an audio jack interface with polarity detection and switching. [IngenicoInc_0138722-IngenicoInc_0138748]

138. Late in Oct'12, Rotsaert presents the Roadmap going forward with Landi including the expected release dates of the various products. This presentation clearly shows a progression

into the future with Landi and BBPOS in the graphic below and refers to "Global strategy: Derivated iTMP/(iTMP+) from from Landi S055P".




***RM1&TR1 ITMP with Landi _ 20121029.pptx [IngenicoInc_0072949]***

139. The sections below describe the specific documents and emails that shared the BBPOS trade secrets with ROAM/Ingenico.

### 6.2.1 Audio Jack Polarity Detection trade secret information requested/received

140. BBPOS shared their design information for detecting the polarity of an audio jack by providing circuit schematics. These circuit schematics identify specific methods for first detecting the polarity of the audio jack and then routing the higher voltage signal to the "Mic input" and routing the lower voltage signal to the "Ground".

- Email: The documents you requested [BBPOS_0005630]

  - Email providing files from BBPOS to ROAM/Ingenico PM per request
  - Sent 7/16/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, Ingenico
  - Attachments:
    - Phone list.xlsx [BBPOS_0005631] – lists the models of phones supported
    - audio interface.pdf [BBPOS_0005632] – schematic showing the **polarity circuit**
    - BBPOS-DataOutputFormat-V1.15.doc [BBPOS_0005633-BBPOS_0005645] – describes the communication formats for specific clients

- Attached Document: audio interface.pdf [BBPOS_0005632]

  - Schematic showing the interface for the Audio Jack devices, including the polarity detection circuitry
  - Sent 7/16/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, Ingenico
  - Attached to email along with multiple files requested by Christopher Rotsaert [BBPOS_0005630]

- Email: Fwd: Paypal G4X – schematic [BBPOS_0005664]

  - Email providing files to BBPOS to ROAM/Ingenico PM per request
  - Sent 7/17/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attachments:
    - Paypal-PCB1-ST04-V3.1.pdf [BBPOS_0005665-BBPOS_0005667] – shows the polarity detection and the power management circuit

- Attached Document: Paypal-PCB1-ST04-V3.1.pdf [BBPOS_0005665-BBPOS_0005667]
  - Schematic showing the design of the PayPal mPOS device including both the polarity detection and the power management circuitry.
  - Sent 7/17/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attached to email sent by Daniel Tsai [BBPOS_0005664]

### 6.2.2 Power Management trade secret information requested/received

141. BBPOS shared their design information for power management in the mPOS device. Power management is tricky in such a small platform and was discussed in many emails

and included in schematics that were shared between BBPOS engineers and ROAM/Ingenico product manager.

- Email: Re: Our confcall next Monday [IngenicoInc_0009883-IngenicoInc_0009891]
  - In this email string, BBPOS engineer Daniel Tsai explains issues with power management in the electronics of the Swiper device. He suggests enlarging the antenna size, adding an RF Amplifier and enlarging the battery for better power management.
  - Sent 5/23/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico PM
  - Attachments: 3 datasheets for chips he is using in his design along with the schematic showing the EMV Swiper electronics design with the power management circuitry [IngenicoInc_0010195-IngenicoInc_0010200]

- Attached Document: EMV_Swiper.pdf [IngenicoInc_0010195-IngenicoInc_0010200]
  - Schematic showing the EMV Swiper electronics design, including the power management circuitry on Pg 5
  - Sent 5/23/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico PM
  - Attached to email along with datasheets for chosen chipsets [IngenicoInc_0009883-IngenicoInc_0009891]

- Email: Re: Our confcall next Monday [IngenicoInc_0134751-IngenicoInc_0134759]
  - In this continued email string, BBPOS engineer Daniel Tsai gives a status of code that exists in one version of the EMV Swiper and that can be ported to the next version. “The currently developed code in cortex M0 which can be ported into M4 are: Audio communication, Magnetic head decoding, USB generic HID, key management, encryption, bootloader.”
  - Sent 5/23/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico PM
  - Attachments: 3 datasheets for chips he is using in his design along with the schematic showing the EMV Swiper electronics design with the power management circuitry, EMV_Swiper.pdf [IngenicoInc_0135063-IngenicoInc_0135068]

- Attached Document: EMV_Swiper.pdf [IngenicoInc_0135063-IngenicoInc_0135068]
  - Schematic showing the EMV Swiper electronics design, including the power management circuitry on Pg 5
  - Sent 5/23/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico PM

- Attached to email along with datasheets for chosen chipsets [IngenicoInc_0134751-IngenicoInc_0134759]

- Email: Fwd: Paypal G4X – schematic [BBPOS_0005664]
  - Email providing files to BBPOS to ROAM/Ingenico PM per request
  - Sent 7/17/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attachments:
    - Paypal-PCB1-ST04-V3.1.pdf [BBPOS_0005665-BBPOS_0005667] – shows the **power management circuitry**

- Attached Document: Paypal-PCB1-ST04-V3.1.pdf [BBPOS_0005665-BBPOS_0005667]
  - Schematic showing the design of the PayPal mPOS device including both the polarity detection and the power management circuitry.
  - Sent 7/17/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attached to email sent by Daniel Tsai [BBPOS_0005664]

- Email: Re: One missing scheme for explanation : solution to handle the 2 categories of phones for amplitude definition [BBPOS_0005646]
  - Email providing explanation of the swiper output with regard to questions on amplitude levels and power management "When the 5KHz tone is replaced by 7KHz tone, the swiper will output in lower amplitude level. Please check attached documents about battery estimation and format ID list."
  - Sent 7/17/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attachments:
    - battery life estimation.pages [BBPOS_0005647-BBPOS_0005648]
    - BBPOS-DataOutputFormat-V1.21.doc [BBPOS_0005649-BBPOS_0005663]

- Email: Re: One missing scheme for explanation : solution to handle the 2 categories of phones for amplitude definition [IngenicoInc_0010655-IngenicoInc_0010656]
  - Related to previous email string – Daniel Tsai BBPOS continues to advise ROAM/Ingenico PM on power management concepts – in this email he is answering specific questions posed by Rotsaert. He actually shares that two components are wired together to implement the capability.
  - Sent 7/27/2012 From Daniel Tsai, BBPOS to Christopher Rotsaert, ROAM/Ingenico

### 6.2.3 Automatic Gain Control (and SDK) trade secret information requested/received

142. BBPOS shared their design information for how they handled Automatic Gain control with the audio communication method used with the audio jack of the mPOS device.

- Email: Re Data rate by audio jack2 [IngenicoInc_0009756-IngenicoInc_0009757]
  - Email where BBPOS describes to the ROAM/Ingenico PM the concept of Automatic Gain Control and how it affects data rates.
  - Sent 4/23/2012 From Ben Lo, BBPOS to Christopher Rotsaert, ROAM/Ingenico per his questions on regarding the data rates across the audio jack.

- Email: RE Visite BBPOS [IngenicoInc_0283863-IngenicoInc_0283864]
  - Email in French mostly where ROAM/Ingenico PM copies the information he received from BBPOS into an email to the Ingenico Dev Team regarding the audio jack sampling rate, AGC and power management.
  - Sent 7/10/2012 From Ben Lo, BBPOS to Christopher Rotsaert, ROAM/Ingenico per his questions regarding the data rates across the audio jack.

### 6.2.4 Communication Formats (and SDK) trade secret information requested/received

143. BBPOS shared their design information for reading data from the mPOS device.

- Email: Re: iWL - android with Roam Data solution [BBPOS_0004382]
  - Email where BBPOS is providing the security algorithm, DUKPT in C++files, and a document about the communication format, specifically output format 10, to ROAM/Ingenico in order to assist Ingenico with developing APIs to the BBPOS developed ROAM swiper.
  - Sent 2/16/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert, ROAM/Ingenico and Jerome Grendemenge, Ingenico R&D
  - Attachments:
    - ROAM Swiper Output Format 10.docx [BBPOS_0004383-BBPOS_0004384]
    - DukptClientTest.zip – comprised of numerous files, refer to list in Data Security / Encryption Methods section below.

- Attached Document: ROAM Swiper Output Format 10.docx [BBPOS_0004383-BBPOS_0004384]
  - Documentation providing information on how to interpret the data coming from the swiper.
  - Sent 2/16/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert, ROAM/Ingenico and Jerome Grendemenge, Ingenico R&D
  - Attached to email sent by Jimmy Tang [BBPOS_0004382]

- Email: Re: iWL - android with Roam Data solution [BBPOS_0004422-BBPOS_0004423]
  - Email where BBPOS is providing the Ingenico Software Architect guidance on the RoamPayAPI with regard to communicating with either the mobile application or the payment server. This appears to be a followup to receiving the security and encryption algorithms but also for interpreting the data communication format received from the mPOS swiper.
  - Sent 2/17/2012 From Jimmy Tang, BBPOS to Jerome Grendemenge, Ingenico Software Architect and Christopher Rotsaert, ROAM/Ingenico PM.

- Email: Re: iWL - android with Roam Data solution [BBPOS_0004622-BBPOS_0004627]
  - Email string where BBPOS engineers and Ingenico engineers are troubleshooting the integration of the Ingenico devices with the mPOS swiper in order to perform a demo. Code, documentation and direction is sent from BBPOS. In part of the email string, Ingenico architect states he has made good progress integrating the code Jimmy Tang, BBPOS provided but still needs additional information. This appears to be a followup to receiving the security and encryption algorithms. Jimmy Tang provides guidance in the emails and another attachment for data communication with the swiper.
  - Sent 2/28/2012 From Jimmy Tang, BBPOS to Jerome Grendemenge, Ingenico Software Architect and Christopher Rotsaert, ROAM/Ingenico PM.
  - Attachment: SwiperAPI-Android-Guide.doc [BBPOS_0004628-BBPOS_0004648]

- Email: Re: <ok [BBPOS_0005112]
  - Email where Jimmy Tang sends to ROAM/Ingenico PM new drafts of designs he is creating for ROAM regarding communication flow to support EMV readers.
  - Sent 7/18/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attachments:
    - BBPOS EMVFlow.docx [BBPOS_0005113-BBPOS_0005114]
    - BBPOS TwoWayCommunication.docx [BBPOS_0005115-BBPOS_0005116]

- Attached Document: BBPOS EMVFlow.docx [BBPOS_0005113-BBPOS_0005114]
  - Documentation on the design of the Communication flow for the newly designed BBPOS EMV readers.
  - Sent 7/18/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert,
  - Attached to email [BBPOS_0005112]

- Attached Document: BBPOS TwoWayCommunication.docx [BBPOS_0005115-BBPOS_0005116]
  - Documentation on the design of the Communication between the reader and the mobile phone for the newly designed BBPOS EMV readers.
  - Sent 7/18/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert,
  - Attached to email [BBPOS_0005112]

- Email: Re: Swiper Track 2 + Track 3 [BBPOS_0005121-BBPOS_0005122]
  - Email where Ben Lo, BBPOS, sends information to ROAM/Ingenico PM about the data communication formats from the readers that are generic with regard to using for the German market.
  - Sent 7/25/2012 From Ben Lo, BBPOS to Christopher Rotsaert, ROAM/Ingenico
  - Attachment: BBPOS-DataOutputFormat-V1.21.doc [BBPOS_0005123-BBPOS_0005137]

- Attached Document: BBPOS-DataOutputFormat-V1.21.doc [BBPOS_0005123-BBPOS_0005137]
  - Documentation on the formats for data communication coming from readers.
  - Sent 7/25/2012 From Ben Lo, BBPOS to Christopher Rotsaert,
  - Attached to email [BBPOS_0005121-BBPOS_0005122]

- Attached Document: SwiperAPI-Android-Guide.doc [BBPOS_0004628-BBPOS_0004648]
  - Document describes how to integrate Swiper functionality into Android applications dated Dec, 2011. It describes the output of the swiper and how to interpret the data received using specific formats identified in the received data stream.
  - Sent 2/28/2012 From Jimmy Tang, BBPOS to Jerome Grendemenge, Ingenico Software Architect and Christopher Rotsaert, ROAM/Ingenico PM.
  - Attached to email [BBPOS_0004622-BBPOS_0004627]

- Email: Format 17 & Format 20 for Track2 [BBPOS_1632236-BBPOS_1632239]
  - Email string where BBPOS provides guidance to ROAM Data on interpreting data from the swiper by using the different data communication formats. The email includes the format parameters and the attachment is a simulator that ROAM developers can use for testing.
  - Sent 4/4/2012 From Ben Lo, BBPOS to ROAM development team.
  - Attachment: SwiperSimulator.exe [BBPOS_1632240]

- Attached Executable: SwiperSimulator.exe [BBPOS_1632240]

- Application to assist engineers developing APIs for BBPOS swiper with interpreting data communication formats.
- Sent 4/4/2012 From Ben Lo, BBPOS to ROAM development team.
- Attached to email [BBPOS_1632236-BBPOS_1632239]

### 6.2.5 Data Security / DUKPT Data Encryption Methods trade secret information requested/received

144. BBPOS shared their design information for detecting the polarity of an audio jack by providing circuit schematics. These circuit schematics identify specific methods for first detecting the polarity of the audio jack and then routing the higher voltage signal to the "Mic input" and routing the lower voltage signal to the "Ground".

- Email: Re: iWL - android with Roam Data solution [BBPOS_0004382]
  - Email where BBPOS is providing the security encryption method, DUKPT algorithm in C++files, and a document about the communication format, specifically output format 10, to ROAM/Ingenico in order to assist Ingenico with developing APIs to the BBPOS developed ROAM swiper.
  - Sent 2/16/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert, ROAM/Ingenico and Jerome Grendemenge, Ingenico Software Architect.
  - Attachments:
    - ROAM Swiper Output Format 10.docx [BBPOS_0004383-BBPOS_0004384]
    - DukptClientTest.zip – comprised of numerous files, refer to list in Data Security / Encryption Methods section below.

- Attached Zip File(s): DukptClientTest.zip
  - This zip file is comprised of many files and file types that were Bates stamped separately. The combination of these files make up source code that would enable the recipient to compile and run this DUKPT version on their own machine and develop interfaces to read and decrypt the data coming from the BBPOS developed ROAM swiper device using this method.
    - DEACore.cpp [BBPOS_0004399-BBPOS_0004406]
    - stdafx.h [BBPOS_0004390]
    - DukptClientTest.cpp [BBPOS_0004413-BBPOS_0004417]
    - DUKPTCore.cpp [BBPOS_0004391-BBPOS_0004397]
    - DEACore.h [BBPOS_0004388]
    - DukptClientTest.vcproj [BBPOS_0004407-BBPOS_0004410]
    - DukptClientTest.sln [BBPOS_0004419]
    - stdafx.cpp [BBPOS_0004389]
    - ReadMe.txt [BBPOS_0004398]
    - DUKPTCore.h [BBPOS_0004418]

    - resource.h [BBPOS_0004412]
    - targetver.h [BBPOS_0004411]
    - DukptClientTest.rc [BBPOS_0004385-BBPOS_0004387]
  - Sent 2/16/2012 From Jimmy Tang, BBPOS to Christopher Rotsaert, ROAM/Ingenico and Jerome Grendemenge, Ingenico R&D
  - Attached to email sent by Jimmy Tang [BBPOS_0004382]

- Email: Re: iWL - android with Roam Data solution [BBPOS_0004422-BBPOS_0004423]
  - Email where BBPOS is providing the Ingenico Software Architect guidance on the RoamPayAPI with regard to communicating with either the mobile application or the payment server. This appears to be a followup to receiving the security and encryption algorithms but also for interpreting the data communication format received from the mPOS swiper.
  - Sent 2/17/2012 From Jimmy Tang, BBPOS to Jerome Grendemenge, Ingenico Software Architect and Christopher Rotsaert, ROAM/Ingenico PM.

- Email: Re: iWL - android with Roam Data solution [BBPOS_0004622-BBPOS_0004627]
  - Email string where BBPOS engineers and Ingenico engineers are troubleshooting the integration of the Ingenico devices with the mPOS swiper in order to perform a demo. Code, documentation and direction is sent from BBPOS. In part of the email string, Ingenico architect states he has made good progress integrating the code Jimmy Tang, BBPOS provided but still needs additional information. This appears to be a followup to receiving the security and encryption algorithms. Jimmy Tang provides guidance in the emails and another attachment.
  - Sent 2/28/2012 From Jimmy Tang, BBPOS to Jerome Grendemenge, Ingenico Software Architect and Christopher Rotsaert, ROAM/Ingenico PM.
  - Attachment: SwiperAPI-Android-Guide.doc [BBPOS_0004628-BBPOS_0004648]

### 6.3 ROAM / Ingenico's use of the Trade Secrets

145. As can be seen in the previous section, ROAM and Ingenico requested and received a significant amount of information from BBPOS specifically regarding BBPOS' 5 asserted trade secrets. The following sections identify Ingenico's use of these trade secrets in the accused Ingenico mPOS devices.

146. The product requirements for the Ingenico devices in, or after 2012, identify the use of BBPOS' trade secrets. Whereas, prior to that, the product requirements for Ingenico devices developed prior to 2011 show no use of the technology identified in the BBPOS trade secrets.

### 6.3.1 Ingenico's use of BBPOS' Audio Jack Polarity Detection design

147. As described above in the Section 5.1.1 BBPOS Audio Jack Polarity Detection Design, BBPOS determined that some mobile phones did not follow the same standard for the Audio Jack microphone and ground connections. And as described above in Section 6.2.1, BBPOS shared this information along with circuit designs with ROAM which were in turn shared with Ingenico. Ingenico also shared BBPOS information with their subsidiary Landi.

148. As a result, this information and circuit design was incorporated into Ingenico / Landi devices including the RP350X, RP750X, the RP100 series, and the RP450 series.

149. The product requirements for the Ingenico devices in, or after 2012, identify the use of BBPOS' design to "detect polarity to switch automatically MIC/GND".

- Product Requirements Document – RP350X version 5.0 (2012)

| | | |
|---|---|---|
| 4.14.5. | RP350x shall detect polarity to switch automatically MIC/GND | M |

***Product Requirement: "detect polarity to switch automatically MIC/GND"***

***PRD RP350X_ v5.0 - 03 12 2012.DOC, 12/3/2012, P20***
***[IngenicoInc_0049942-IngenicoInc_0049966]***

- Product Requirements Document – RP750X version 4.0 (2013)

| | | |
|---|---|---|
| 4.15.3. | RP750x shall detect polarity to switch automatically MIC/GND | M |

***Product Requirement: "detect polarity to switch automatically MIC/GND"***

***PRD RP750X_ v4.0 - 31 01 2013.DOC, 2/1/2013, P28***
***[IngenicoInc_0158490-IngenicoInc_0158525]***

- Product Requirements Document – RP100X Draft (2013)

| | | |
|---|---|---|
| 3.8.3. | RP100x shall detect polarity to switch automatically MIC/GND | |

***Product Requirement: "detect polarity to switch automatically MIC/GND"***

***PRD RP100x DRAFT.DOC, 5/24/2013, P10 [IngenicoInc_0190250-IngenicoInc_0190265]***

- Product Requirements Document - RP150X Version 2.0 (2013)

| | | |
|---|---|---|
| 4.14.5. | RP150x shall detect polarity to switch automatically MIC/GND | M |

***Product Requirement: "detect polarity to switch automatically MIC/GND"***

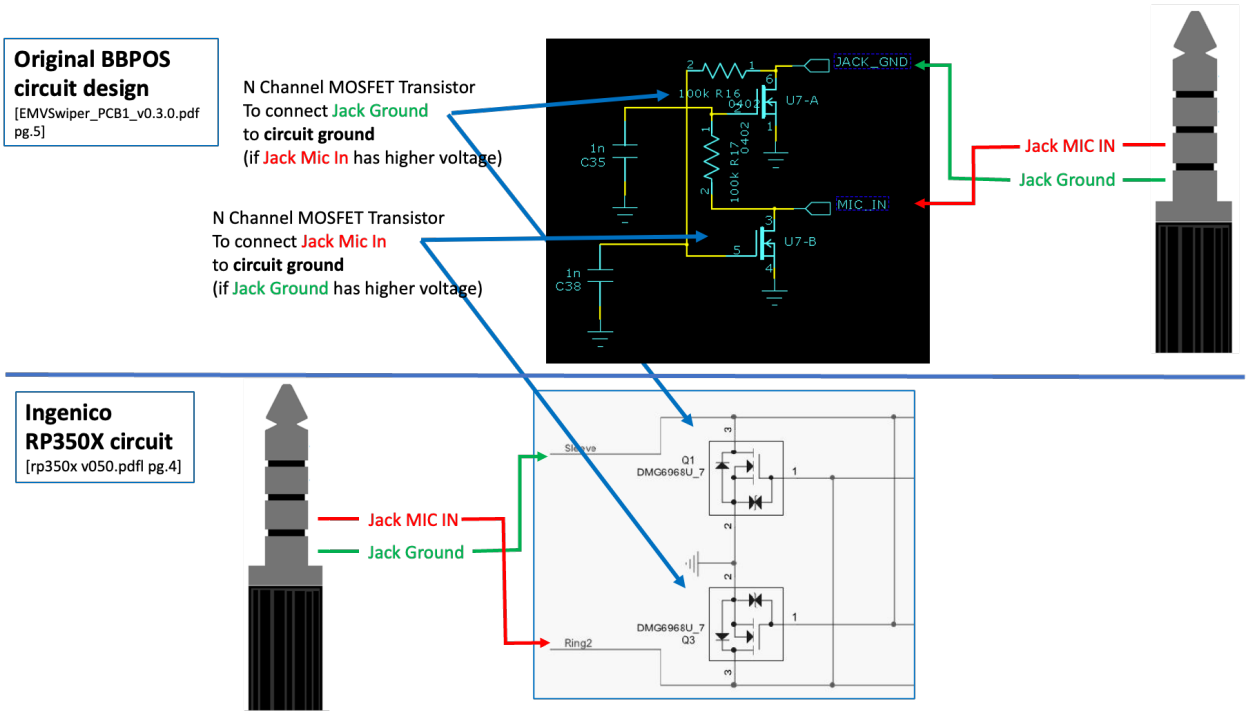
***PRD RP150X _ v2.0 - 07 01 2013.DOC, 1/7/2013, P17 [IngenicoInc_0076359-IngenicoInc_0076380]***

150. The following subsections show additional detail as to how the BBPOS Polarity Detection design was incorporated into these Ingenico devices.

#### *6.3.1.1 Ingenico RP350X uses BBPOS' design for Polarity detection and reversal*

151. The RP350X is the first mPOS device that Ingenico/Landi released for sale. The RP350X served as a platform or basis for subsequent Ingenico mPOS devices including the RP750X, RP100 series, and the RP450 series.

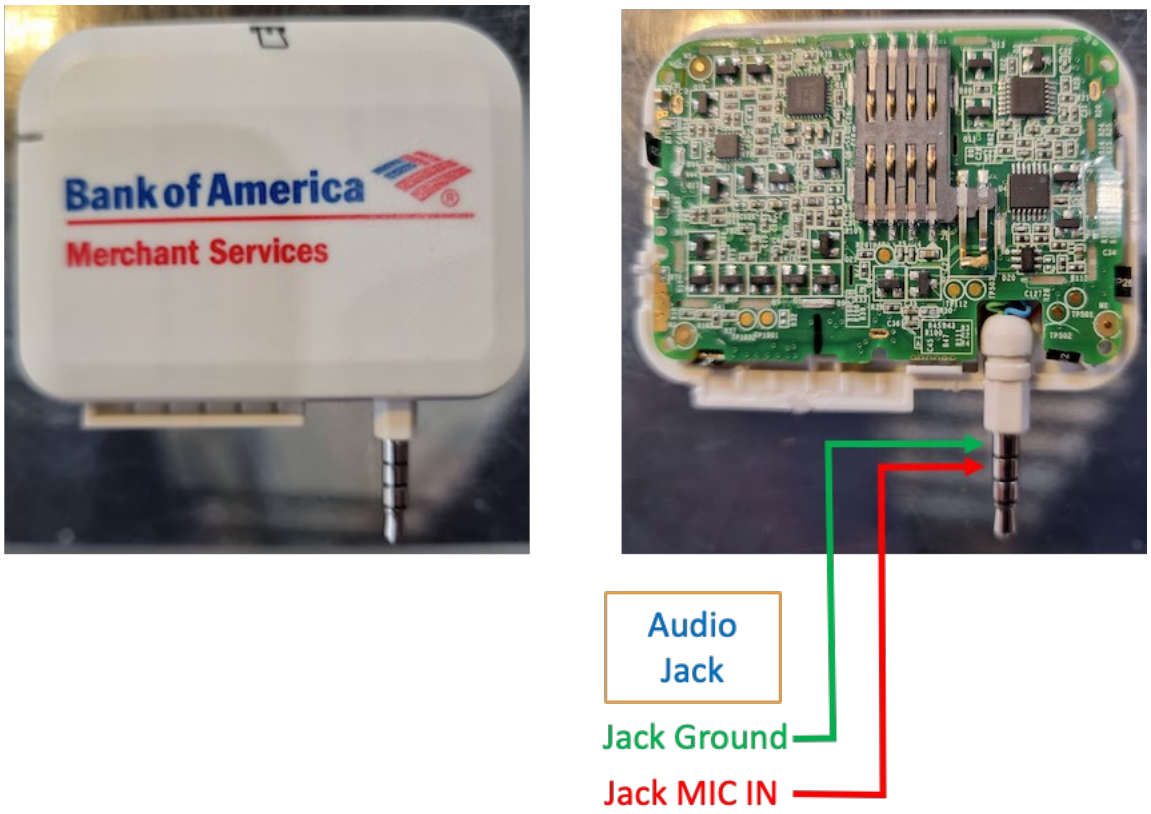
152. In the RP350X device, Ingenico used the BBPOS circuit design that incorporated a single pair of N channel MOSFET transistors to determine if the Jack MIC IN or the Jack Ground has the higher voltage, and to connect the correct signal to the **circuit ground** of the mPOS device. This is shown below in the comparison of BBPOS' original circuit schematic and Ingenico's RP350X circuit schematic.




***Ingenico's RP350X using same circuit design as BBPOS' Polarity detection and reversal design***

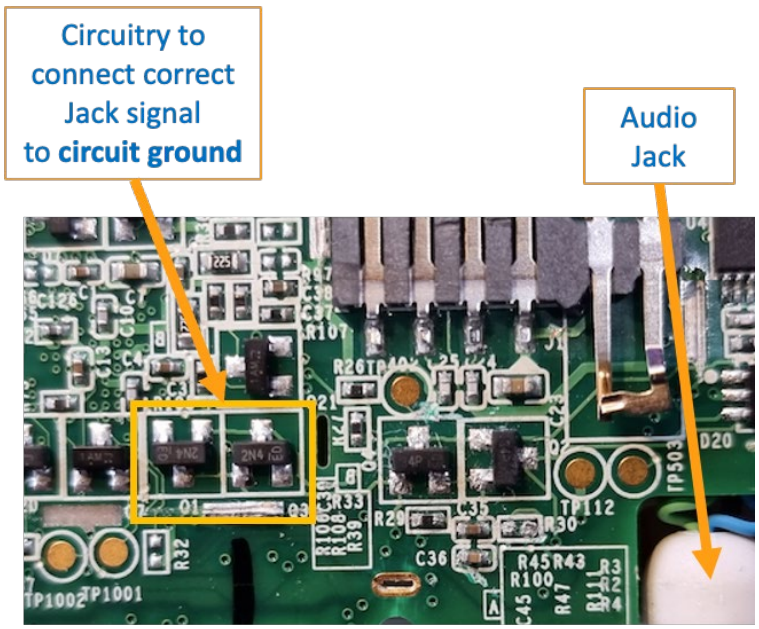
***153.*** The use of this BBPOS design in an Ingenico device is sufficient to both detect and reverse the polarity of the Jack signals, by connecting the correct Jack signal to the mPOS circuit ground. This automatically ensures the correct polarity for both the Jack Ground and the Jack MIC IN signals for the mPOS device.

154. The following shows the physical Ingenico RP350X mPOS device and the internal circuitry.




***Ingenico RP350X mPOS device and internal circuitry***

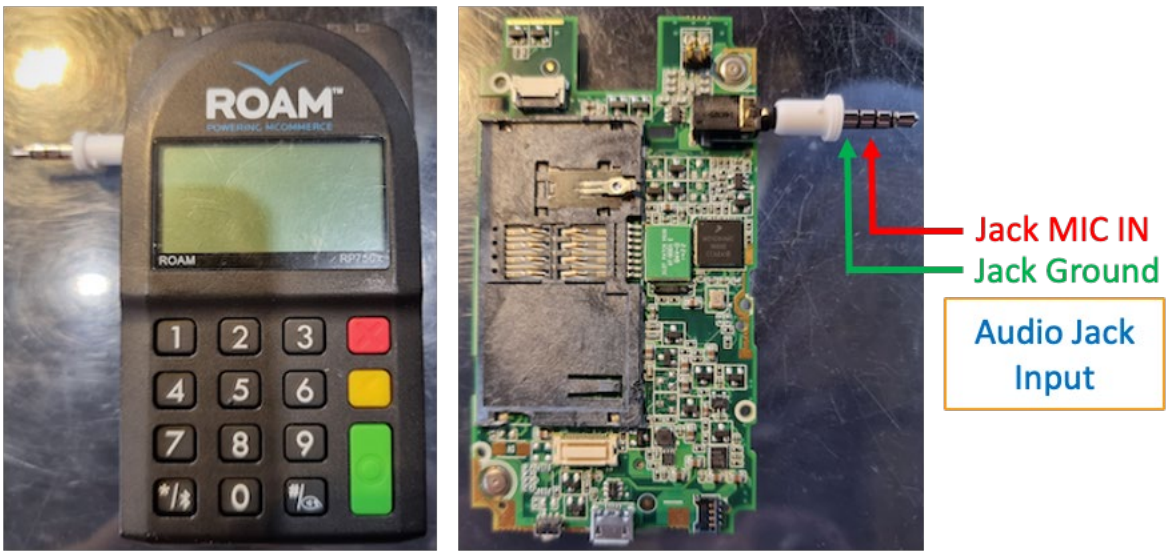
155. The following shows the specific circuitry that performs the Polarity detection and reversal for the circuit ground. This uses the same design and concepts that BBPOS provided to ROAM that were shared with Ingenico.




***Ingenico RP350X circuitry that performs BBPOS' polarity detection and reversal***

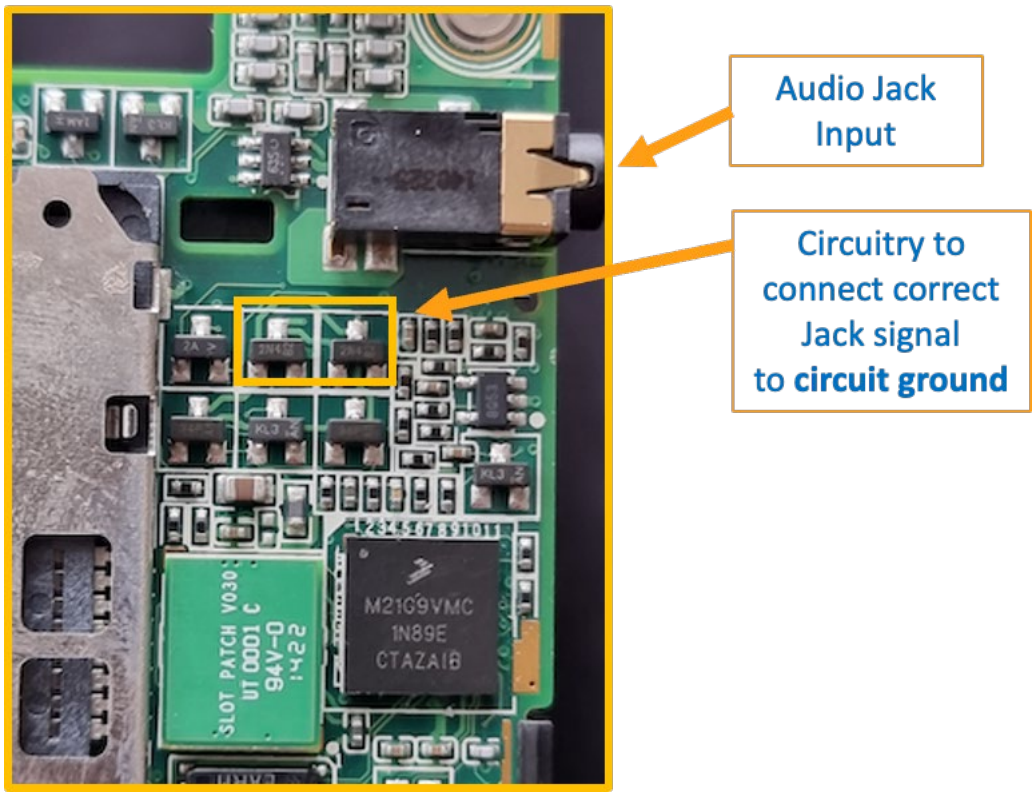
#### *6.3.1.2 Ingenico RP750X uses BBPOS' design for Polarity detection and reversal*

156. The following shows the physical Ingenico RP750X mPOS device and the internal circuitry. The device I tested below is ROAM's prototype for the Ingenico RP750X device.




***Ingenico RP750X mPOS device and internal circuitry***

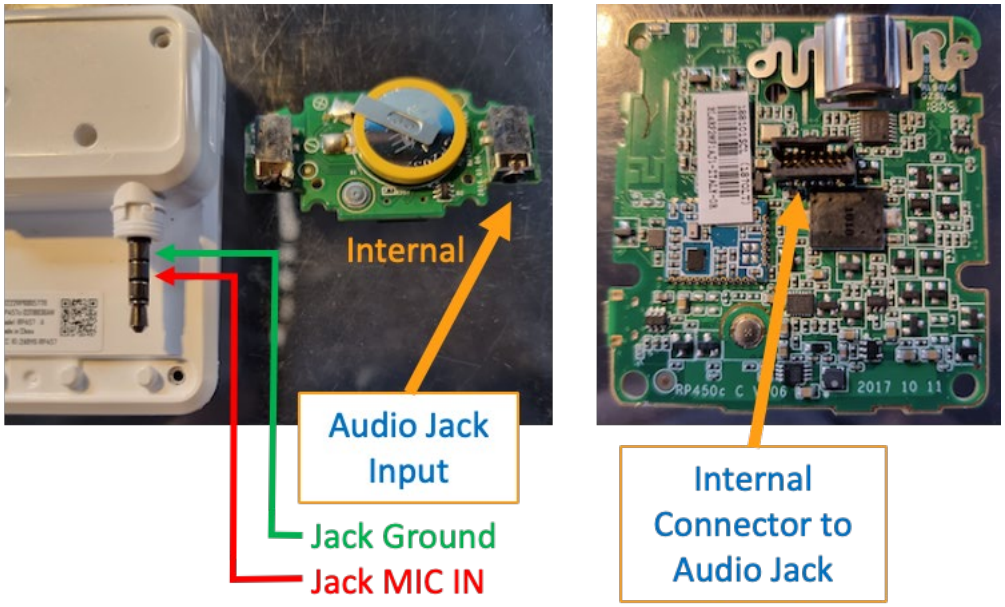
157. The following shows the specific circuitry that performs the Polarity detection and reversal for both the circuit ground and the circuit microphone. This uses the same design and concepts that BBPOS provided to ROAM that were shared with Ingenico.




***Ingenico RP750X circuitry that performs BBPOS' polarity detection and reversal***

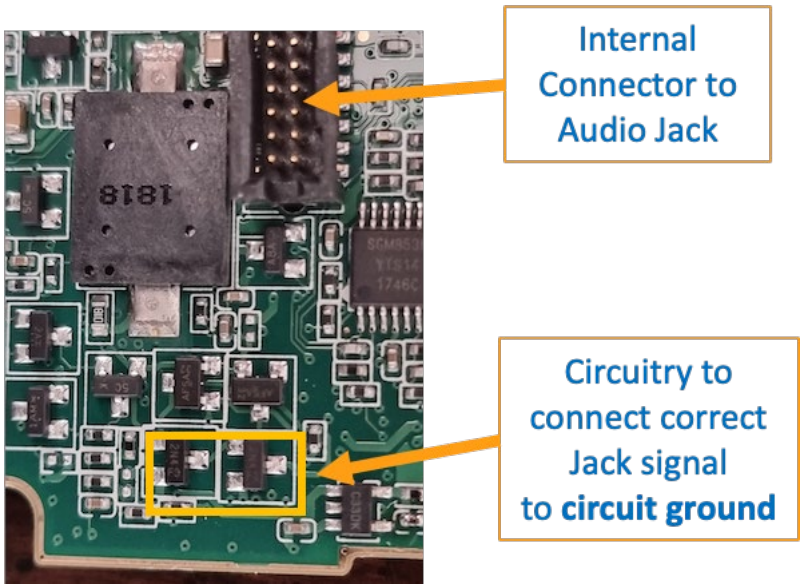
#### *6.3.1.3 Ingenico RP457C uses BBPOS' design for Polarity detection and reversal*

158. The following shows the physical Ingenico RP457C mPOS device and the internal circuitry.




***Ingenico RP457C mPOS device and internal circuitry***

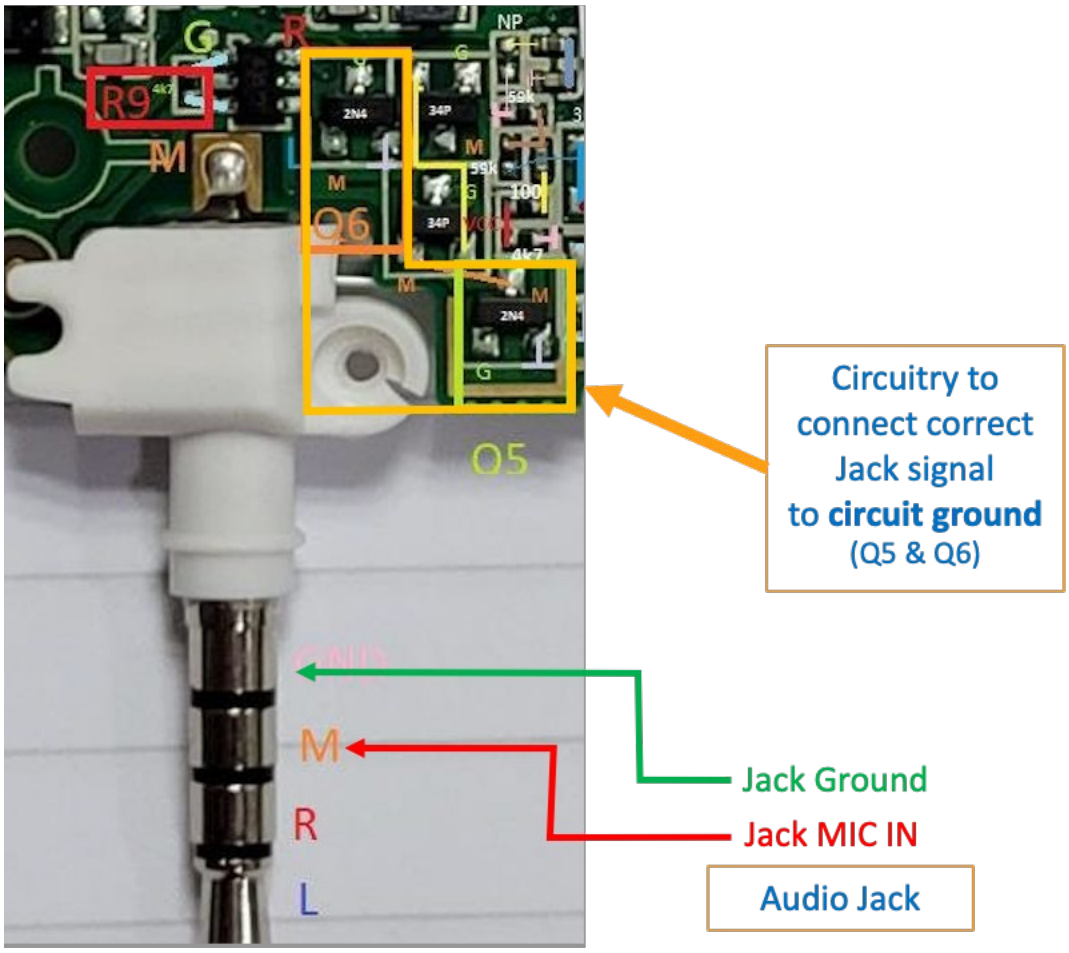
159. The following shows the specific circuitry that performs the Polarity detection and reversal for both the circuit ground and the circuit microphone. This uses the same design and concepts that BBPOS provided to ROAM that were shared with Ingenico.




***Ingenico RP457C circuitry that performs BBPOS' polarity detection and reversal***

#### *6.3.1.4 Ingenico RP170C uses BBPOS' design for Polarity detection and reversal*

160. The following shows the specific circuitry that performs the Polarity detection and reversal for both the circuit ground and the circuit microphone. This uses the same design and concepts that BBPOS provided to ROAM that were shared with Ingenico.




***Ingenico RP170C circuitry that performs BBPOS' polarity detection and reversal***

### 6.3.2 Ingenico's use of BBPOS's Power Management design (Auto Power On)

161. As mentioned in Section 5.1.2, BBPOS' Power Management Design allows the mPOS to automatically power on when the mobile phone is plugged in and the audio jack

interface is active. Also described above in Section 6.2.2, BBPOS provided their designs for power management to Ingenico, specifically including automatic power on designs.

162. The product requirements for the Ingenico devices in 2013 cited below identify the use of BBPOS' design to "automatically power on when plugged on Mobile".

- Product Requirements Document – RP350X version 5.0 (2012)

| | | |
|---|---|---|
| 4.21.7. | RP350x shall automatically power on when plugged on Mobile | M |
| 4.21.8. | RP350x shall automatically power off when unplugged on Mobile | M |

***Product Requirement: "automatically power on when plugged on Mobile"***

***PRD RP350X _ v5.0 - 03 12 2012.DOC, 12/3/2012, P22***
***[IngenicoInc_0049942-IngenicoInc_0049966]***

- Product Requirements Document – RP750X version 7.0 (2013)

| | | |
|---|---|---|
| 4.5.10. | Idle screen<br>When device is power on (or wake-up) out of transaction operation, RP750x should display an idle screen based on a logo which may be customized (only for specific product with customer industrial design)<br>ROAM | M |

***Product Requirement: "device is power on (or wake-up) out of transaction operation"***

***PRD RP750X _ v7.0 - 04 03 2013.DOC, 3/4/2013, P25***
***[IngenicoInc_0181636-IngenicoInc_0181675]***

- Product Requirements Document - RP150X Version 2.0 (2013)

| | | |
|---|---|---|
| 4.21.6. | RP150x shall automatically power on when plugged on Mobile | M |
| 4.21.7. | RP150x shall automatically power off when unplugged on Mobile | M |

***Product Requirement: "automatically power on when plugged on Mobile"***

***PRD RP150X _ v2.0 - 07 01 2013.DOC, 1/7/2013, P20***
***[IngenicoInc_0076359-IngenicoInc_0076380]***

- Product Requirements Document - RP100X DRAFT (2013)

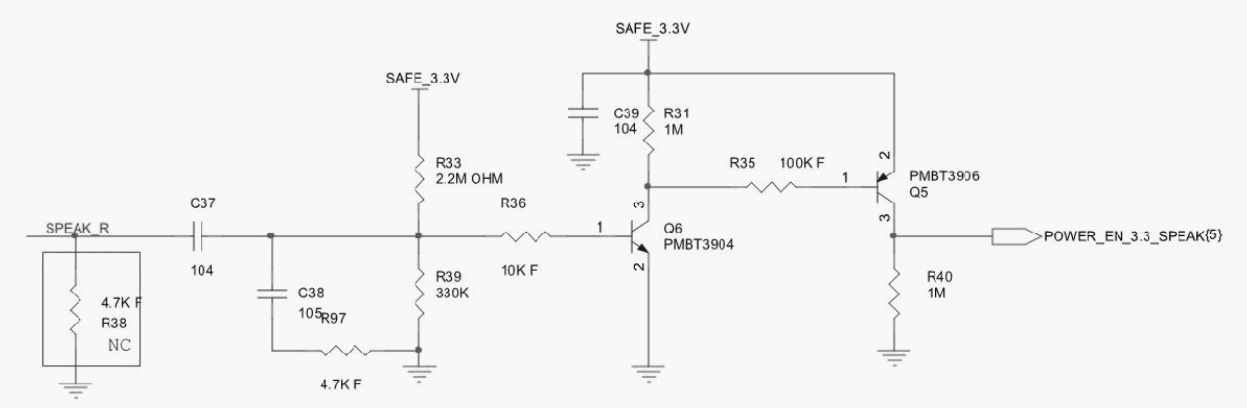
| | | |
|---|---|---|
| 3.21.5. | RP100x shall automatically power off when unplugged on Mobile | M |

***PRD RP100x DRAFT.DOC, 5/24/2013, P14 [IngenicoInc_0190250-IngenicoInc_0190265]***

163. The specific behavior of the Ingenico devices is that, once the device is asleep, it will wake-up when the mobile phone is plugged in and there is valid activity from the mobile phone on the audio jack interface. Once the device is awake it will determine if the signal from the audio jack interface is valid (e.g. the mobile phone payment app requests an initialization or transaction from the mPOS device). If it is not a valid mPOS signal the power does not remain on.

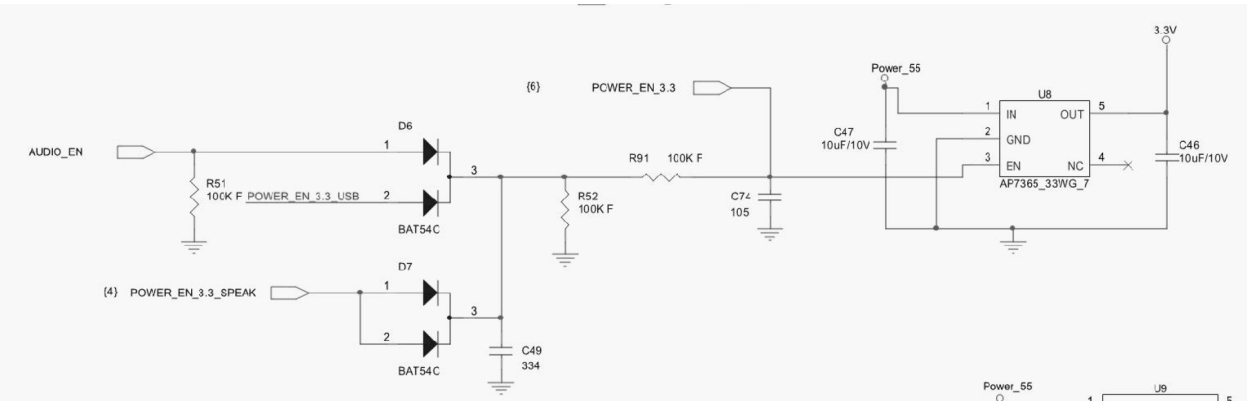
164. I tested this behavior of the RP350X, RP750X, and the RP457C devices by monitoring the activity on the audio jack interface with an oscilloscope. For example, simply plugging in the audio jack cable from the mobile phone to the mPOS device was not sufficient to power on the device. When I played music on the mobile phone through the audio jack interface, this was not sufficient to have the Ingenico device remain on. However, I then started a test transaction from the mobile phone application. The test transaction begins with a device initialization over the audio jack interface in order to determine the type of mPOS device. When that mPOS initialization occurs, the Ingenico device turns-on and remains on, exhibiting the same behavior as the BBPOS design.

165. In addition, the schematics for the Ingenico RP350X contain the same type of trigger circuit and power switch as that of the BBPOS design. The following shows the equivalent of BBPOS' "temporary trigger" that uses the input from the SPEAK_R signal (right audio channel), on the left side of the diagram, to produce a trigger signal POWER_EN_3.3_SPEAK on the right side.




*rp350x v050.pdf pg.4 [IngenicoInc_0283923-IngenicoInc_0283931]*

166. That POWER_EN_3.3_SPEAK trigger signal, now shown below at the lower left side, is then input to the 'enable' pin on the "power switch" on the right side (IC component AP7365_33WG_7). That "power switch" will then supply power to the microprocessor on the mPOS device via the 3.3v (i.e. 3.3 volts applied to the VDD of the microprocessor).




*rp350x v050.pdf pg.5 [IngenicoInc_0283923-IngenicoInc_0283931]*

167. The power switch in the schematic above can also receive trigger signals from other sources such as the AUDIO_EN signal in the top right side of the above diagram that is enabled by the microprocessor, which I believe acts in a similar fashion to a "permanent trigger" of the BBPOS design.

### 6.3.3 Ingenico's use of BBPOS' Pre-analyzed communication settings and adaptive threshold (or Auto Gain Control)

168. As discussed in Section 5.1.3, one of BBPOS' trade secrets is the information that they produced as a result of analyzing and determining the audio and communication characteristics of over 442 mobile phone models. Section 6.2.3 above, also describes the information BBPOS shared with Ingenico regarding their designs for communication settings and Automatic Gain Control (AGC).

169. This information allowed the configuration of the communication settings of the mobile phone to achieve the best communication results. The parameters of the communication settings for the BBPOS SDK included:

- the frequency or speed of the incoming (receive) transmissions [***MaxInputFrequency]***
- the frequency or speed of the outgoing (send) transmissions [***MaxOutputFrequency]***
- the volume setting of the audio jack interface [***LowVolume***]
- the size of data objects being transmitted [***e.g. PreambleLength***]
- the audio source type [***MicrophoneAudioSource***]

170. I was able to compare the parameters for the communication settings defined by BBPOS with the parameters of the communication settings used by Ingenico for the Landi mPOS devices. To do this I retrieved a copy of the SDK used for the PayPal branded mPOS devices (which included both the Ingenico/Landi and BBPOS devices). This is available on the public web site at:

https://mvnrepository.com/artifact/com.PayPal.retail/here-sdk-debug/2.1.02.19063010

171. I was also able to "decompile" this SDK and determined that it contains, among other software components, the "ROAM unified API". As part of the ROAM unified API were two other subcomponents labeled as the BBPOS SDK and the Landicorp SDK.

172. From the decompiled source code of the PayPal SDK, I was able to determine that the Landicorp SDK utilized many of the same parameters for the mobile phone communication settings that BBPOS had established. I identified communication settings for 272 mobile phone models within the Landicorp SDK. The parameters of these communication settings for the Ingenico/Landi SDK included:

- the frequency or speed of the incoming (receive) transmissions [***RecvBaud***]
- the frequency or speed of the outgoing (send) transmissions [***SendBaud***]
- the volume setting of the audio jack interface [***SendVolume***]
- the size of data objects being transmitted [***FrameLength***]
- the audio source type [***AudioSource***]

173. The similar parameter settings used by both companies indicate that BBPOS devices and Ingenico devices optimized their audio interfaces to communicate data at high speeds (over 3000bps / baud). This would have required the use of adaptive threshold methods (i.e. Auto Gain Control) that BBPOS had developed and shared with Ingenico. Therefore, it is my opinion Ingenico utilized BBPOS' Auto Gain Control trade secret information in their product designs.

### 6.3.4 Ingenico's use of BBPOS' Communication Formats

174. As mentioned in Section 5.1.4, BBPOS' proprietary mPOS communication formats were developed after working extensively with financial and payment processing companies that developing mobile payment applications. Section 6.2.4 describes the information BBPOS shared with Ingenico regarding their Communication Formats.

175. The following indicates that Ingenico utilized this information in their product design. For example, the following reference in Ingenico's requirements document for a Key Management System identifies the use of at least two of those communication formats (format 11 and 29):

High level description :

- 3rd Party partner for BDK (batch of several BDK) generation & strorage
- No more BDK created by BBPOS for new P/N, replacement of BDK for Roam production standard. New P/N with Data DUKPT format (format 11 & equivalent of 29 for Data DUKPT)
- Key custodians for transmission to Roam Datacenter & Landi/BBPOS factory
- Key injection in Flextronics with TR39 compliant Landi solution (key loader, secure room) with Landi
- Decryption made in HSM without redundancy
- Decryption / HSM packaged to be delivered as an appliance for installation into a 3rd party datacenter (with MCM5 license)

***PRD Key Management System _ v1.0.2.DOC, pg. 10 [IngenicoInc_0079958-IngenicoInc_0079977]***

### 6.3.5 Ingenico's use of BBPOS Data Security / Encryption Methods (DUKPT data method)

176. As mentioned in Section 5.1.5, BBPOS' proprietary data encryption utilizes a method that derives a DUKPT data key using the standard data variant without performing the TDES self-encryption of the result of the applied Data variants. Section 6.2.5 describes the information BBPOS shared with Ingenico regarding their DUKPT data encryption methods.

177. Ingenico utilized this information in their product designs. For example, Ingenico identifies in their requirements document for a Key Management System that they will utilize the BBPOS' "Data DUKPT" method that is specified in BBPOS' communication format 11 and format 29.

High level description :

- 3rd Party partner for BDK (batch of several BDK) generation & strorage
- No more BDK created by BBPOS for new P/N, replacement of BDK for Roam production standard. New P/N with Data DUKPT format (format 11 & equivalent of 29 for Data DUKPT)
- Key custodians for transmission to Roam Datacenter & Landi/BBPOS factory
- Key injection in Flextronics with TR39 compliant Landi solution (key loader, secure room) with Landi
- Decryption made in HSM without redundancy
- Decryption / HSM packaged to be delivered as an appliance for installation into a 3rd party datacenter (with MCM5 license)

***PRD Key Management System _ v1.0.2.DOC, pg. 10 [IngenicoInc_0079958-IngenicoInc_0079977]***

178. Later in the same document, Ingenico also mentions the use of the BDK (Base Derivative Key) formation after the migration of the Landi products to the "Data DUKPT" method.

| | | |
|---|---|---|
| 4.3.6. | Same BDK format for Landi product & BBPOS (after migration to Data DUKPT) | W |

***PRD Key Management System _ v1.0.2.DOC, pg. 16 [IngenicoInc_0079958-IngenicoInc_0079977]***

179. Again, in the same document, Ingenico identifies that the use of BBPOS Encryption should be "supported in both their Test & Production" products as a Service.

| | | |
|---|---|---|
| 4.6.6. | BBPOS Encryption should be supported in both Test & Production as a Service | M |
| 4.6.7. | BBPOS Encryption should be available as a License model | M |

***PRD Key Management System _ v1.0.2.DOC, pg. 19 [IngenicoInc_0079958-IngenicoInc_0079977]***

## 7 SUMMARY OF OPINIONS

180. My opinions in this report are based on my knowledge, education, experience, and review of the various documents and other information referenced in my report and the attached exhibits. My opinions regarding this case and related issues, as well as my bases and support for those opinions, are set forth in detail throughout this report and are not limited to the opinions summarized below. I expect to testify regarding Defendant's products and BBPOS' products, including whether Defendant's products were misused and/or misappropriated in their own products. I also expect to testify regarding technical background information relevant to the issues in this case as set forth herein.

### 7.1 BBPOS' proprietary product

181. As described in Section 5, BBPOS has developed a number of unique and proprietary features and functions within their mPOS devices. The totality of these unique and proprietary functions form a sophisticated and complex product that required a significant amount of time, effort and financial investment to develop. No one else, either within the industry or outside the industry, had developed or would be likely develop these specific proprietary functions without the benefit of confidential information obtained from BBPOS.

### 7.2 Ingenico's use of the proprietary information

182. BBPOS Polarity Detection designs: ROAM and Ingenico requested and readily received trade secret information from BBPOS regarding Polarity detection. ROAM and Ingenico also discussed and questioned BBPOS for further details on the concepts and operation of their Polarity detection design. The Ingenico devices that were produced subsequent to receiving that trade secret information contained the same concepts and design as the BBPOS Polarity detection designs.

183. This capability was introduced by Ingenico/Landi when producing the RP350X models and future iTMP models which came after their engagement with BBPOS/ROAM Data development. Prior to the RP350X requirements and specification documents which appeared in late 2012, as part of the Landi SOW, polarity detection had not been addressed. The RP350X document shows a "Must Have" requirement of "RP350x shall detect polarity to switch automatically MIC/GND". These can be seen in the requirements documents for the RP350x, the RP750X, the RP100 series and the RP450 series of products:

- PRD RP350X _ v5.0 - 03 12 2012.DOC, 12/3/2012, P20 [IngenicoInc_0049942-IngenicoInc_0049966]

- PRD RP750X _ v4.0 - 31 01 2013.DOC, 2/1/2013, P28 [IngenicoInc_0158490-IngenicoInc_0158525]
- PRD RP100x DRAFT.DOC, 5/24/2013, P10 [IngenicoInc_0190250-IngenicoInc_0190265]
- PRD RP150X _ v2.0 - 07 01 2013.DOC, 1/7/2013, P17 [IngenicoInc_0076359-IngenicoInc_0076380]

184. BBPOS Power Management designs: ROAM and Ingenico requested and readily received trade secret information from BBPOS regarding Power Management. ROAM and Ingenico also discussed and questioned BBPOS for further details on the concepts and operation of battery usage and power management. The Ingenico devices that were produced subsequent to receiving that trade secret information contained the same concepts and design as the BBPOS power management design.

185. This capability was introduced by Ingenico/Landi when producing the RP350X models and future iTMP models which came after their engagement with BBPOS/ROAM Data development. The RP350X document shows a "Must Have" requirement of "RP350x shall automatically power on when plugged on Mobile" and "RP350x shall automatically power off when unplugged on Mobile". These can be seen in the requirements documents for the RP350x, the RP750X, the RP100 series and the RP450 series of products:

- PRD RP350X _ v5.0 - 03 12 2012.DOC, 12/3/2012, P22 [IngenicoInc_0049942-IngenicoInc_0049966]
- PRD RP750X _ v7.0 - 04 03 2013.DOC, 3/4/2013, P25 [IngenicoInc_0181636-IngenicoInc_0181675]

- PRD RP100x DRAFT.DOC, 5/24/2013, P14 [IngenicoInc_0190250-IngenicoInc_0190265]
- PRD RP150X _ v2.0 - 07 01 2013.DOC, 1/7/2013, P20 [IngenicoInc_0076359-IngenicoInc_0076380]

186. BBPOS Automatic Gain Control designs: ROAM and Ingenico requested and readily received trade secret information from BBPOS regarding Power Management. ROAM and Ingenico also discussed and questioned BBPOS for further details on the concepts and operation of battery usage and power management. The Ingenico devices that were produced subsequent to receiving that trade secret information contained the same concepts and design as the BBPOS power management design.

- This capability was introduced by Ingenico/Landi when producing the RP350X models and future iTMP models which came after their engagement with BBPOS/ROAM Data development. The similar parameter settings used by both companies indicate that BBPOS devices and Ingenico devices optimized their audio interfaces to communicate data at high speeds (over 3000bps / baud). This would have required the use of adaptive threshold methods (i.e. Auto Gain Control) that BBPOS had developed and shared with Ingenico. The PayPal SDK Here application, for example, shows the use of these same parameters and is available at: https://mvnrepository.com/artifact/com.PayPal.retail/here-sdk-debug/2.1.02.19063010

187. BBPOS Communication Interface designs: ROAM and Ingenico requested and readily received trade secret information from BBPOS regarding the Communication Interface formats that BBPOS painstakingly worked through by

testing and configuring their software according to the various payment processing providers, such as PayPal.

- This functionality was used by Ingenico/Landi when producing the RP350X models and future iTMP models which came after their engagement with BBPOS/ROAM Data development. The Requirements document for the Key Management System that was used in all iTMP products refers to the use of a set of these formats. Reference document: PRD Key Management System _ v1.0.2.DOC, pg. 10 [IngenicoInc_0079958-IngenicoInc_0079977]

188. BBPOS Data Security design: ROAM and Ingenico requested and readily received trade secret information from BBPOS regarding Data Security, specifically the nature and use of the DUKPT algorithms for data encryption instead of only PIN encryption.

- This functionality was used by Ingenico/Landi when producing the RP350X models and future iTMP models which came after their engagement with BBPOS/ROAM Data development. The Requirements document for the Key Management System that was used in all iTMP products refers to the use of Data DUKPT algorithm. Specifically, this document refers to "Same BDK format for Landi product & BBPOS (after migration to Data DUKPT). BBPOS shared how they used the Data DUKPT algorithm for encrypting the card data which was innovative at the time. Previously, DUKPT had only been used for PIN encryption.
  - PRD Key Management System _ v1.0.2.DOC, pp. 10 & 16 [IngenicoInc_0079958-IngenicoInc_0079977]

189. As described in Section 7, it is my opinion that Ingenico used BBPOS' proprietary information to develop its competing product. Ingenico also incorporated the proprietary features within a short period time; almost in parallel and/or within months, after being engaged with BBPOS and ROAM on the PayPal project (May'12) and Cartes Demo effort (Nov'12).

190. I have analyzed and compared the similarities between the Ingenico iTMP products and the BBPOS products. I have demonstrated that the similarities in Ingenico's Polarity Detection, Power Management, Automatic Gain Control, Encryption, Data Communication Format features are based either wholly or in part on BBPOS' proprietary information.

191. Based on my analysis, it is my opinion that no matter how much effort and capital Ingenico invested, they could not have produced their product within that time period without the use of BBPOS' Proprietary information.

192. The following table summarizes my opinions about the trade secrets used by Ingenico with the associated accused products:

### 7.2.1 TABLE SUMMARIZING EXPERT OPINION REGARDING THEFT OF BBPOS'S TRADE SECRETS

| No. | BBPOS Trade Secret | Brief Description | Unauthorized Recipient | Improper Means | Disclosure / Usage / Expression |
|---|---|---|---|---|---|
| 1 | **Audio Jack Polarity Detection** | *This trade secret determines if the base of a mobile phone's audio jack has a positive or negative polarity and how to route the microphone / input signal appropriately. This enables a single solution to support multiple mobile phone signal formats.* | 1. Ingenico | Primarily, from 2/2012 through 8/2012, at the request and direction of ROAM under the BBPOS-ROAM Licensing Agreement, BBPOS BBPOS transmitted proprietary information in the form of schematics, design documents, source code, etc., among other protected, confidential information of BBPOS, on many occasions to ROAM / Ingenico.[6] | Various improper disclosures by ROAM / Ingenico, including, without limitation, to Landi. |
| | | | 2. Landi | Contemporaneously, Ingenico (via Mr. Rotsaert and/or at his or other Ingenico's executives' directive(s)) directly discloses such trade secrets and other protected, confidential information to Landi. | ***Accused Devices:*** RP350X, RP750X, RP100 series and RP450 series |
| 2 | **Power Management** | *This trade secret provides methods for efficient power use for battery powered mPOS devices as well as performing sleep and auto wakeup (Power on) functions in order to conserve power.* | 1. Ingenico | Primarily, from 2/2012 through 8/2012, at the request and direction of ROAM under the BBPOS-ROAM Licensing Agreement, BBPOS BBPOS transmitted proprietary information in the form of schematics, design documents, source code, etc., among other protected, confidential information of BBPOS, on | Various improper disclosures by ROAM / Ingenico, including, without limitation, to Landi. |

[6] [*See, e.g.,* BBPOS_0005630; BBPOS_0005631; BBPOS_0005632; BBPOS_0005633-BBPOS_0005645; BBPOS_0005632; BBPOS_0005630; BBPOS_0005664; BBPOS_0005665-BBPOS_0005667; BBPOS_0005665-BBPOS_0005667; BBPOS_0005664, *etc*.]

| | | | | | |
|---|---|---|---|---|---|
| | | | | many occasions to ROAM / Ingenico.[7] | |
| | | | 2. Landi | Contemporaneously, Ingenico (via Mr. Rotsaert and/or at his or other Ingenico's executives' directive(s)) directly discloses such trade secrets and other protected, confidential information to Landi. | ***Accused Devices:*** RP350X, RP750X, RP100 series and RP450 series |
| **3** | **Signal Control Settings and Automatic Gain Control** | *This trade secret determines the appropriate gain (e.g., signal thresholds) to use in decoding data, and determining at what speed to reliably transmit and receive the information based parameters defined for the specific mobile phone being used.* | 1. Ingenico | From 2/2012 through 8/2012, at the request and direction of ROAM under the BBPOS-ROAM Licensing Agreement, BBPOS discloses such trade secrets and other protected, confidential information of BBPOS to ROAM / Ingenico.[8] | Various improper disclosures by ROAM / Ingenico, including, without limitation, to Landi. |
| | | | 2. Landi | Contemporaneously, Ingenico (via Mr. Rotsaert and/or at his or other Ingenico's executives' directive(s)) directly discloses such trade secrets and other protected, confidential information to Landi. | ***Accused Devices:*** RP350X, RP750X, RP100 series and RP450 series |
| **4** | **Communication Formats** | *This trade secret provides over 25 different formats for sending credit card and transaction related information between the mPOS device and the mobile phone to ensure compatibility with different mobile payment vendor applications.* | 1. Ingenico | Primarily, from 2/2012 through 8/2012, at the request and direction of ROAM under the BBPOS-ROAM Licensing Agreement, BBPOS BBPOS transmitted proprietary information in the form of schematics, design documents, source code, etc., among other protected, confidential information of BBPOS, on | Various improper disclosures by ROAM / Ingenico, including, without limitation, to Landi. |

[7] [*See, e.g.,* IngenicoInc_0009883-IngenicoInc_0009891; IngenicoInc_0010195-IngenicoInc_0010200; IngenicoInc_0010195-IngenicoInc_0010200; IngenicoInc_0009883-IngenicoInc_0009891; IngenicoInc_0135063-IngenicoInc_0135068; IngenicoInc_0135063-IngenicoInc_0135068; IngenicoInc_0134751-IngenicoInc_0134759; BBPOS_0005664; BBPOS_0005665-BBPOS_0005667; BBPOS_0005665-BBPOS_0005667; BBPOS_0005664; BBPOS_0005646; BBPOS_0005647-BBPOS_0005648; BBPOS_0005649-BBPOS_0005663; IngenicoInc_0010655-IngenicoInc_0010656, *etc.*]

[8] [*See, e.g.,* IngenicoInc_0009756-IngenicoInc_0009757; IngenicoInc_0283863-IngenicoInc_0283864, *etc.*]

| | | | | | |
|---|---|---|---|---|---|
| | | | | many occasions to ROAM / Ingenico.[9] | |
| | | | 2. Landi | Contemporaneously, Ingenico (via Mr. Rotsaert and/or at his or other Ingenico's executives' directive(s)) directly discloses such trade secrets and other protected, confidential information to Landi. | ***Accused Devices:*** RP350X, RP750X, RP100 series and RP450 series |
| **5** | **Data Security / Encryption Methods** | *This trade secret provides methods for encrypting credit card data based on variations of data encryption methods.* | 1. Ingenico | Primarily, from 2/2012 through 8/2012, at the request and direction of ROAM under the BBPOS-ROAM Licensing Agreement, BBPOS BBPOS transmitted proprietary information in the form of schematics, design documents, source code, etc., among other protected, confidential information of BBPOS, on many occasions to ROAM / Ingenico.[10] | Various improper disclosures by ROAM / Ingenico, including, without limitation, to Landi. |
| | | | 2. Landi | Contemporaneously, Ingenico (via Mr. Rotsaert and/or at his or other Ingenico's executives' directive(s)) directly discloses such trade secrets and other protected, confidential information to Landi. | ***Accused Devices:*** RP350X, RP750X, RP100 series and RP450 series |

[9] [*See, e.g.,* BBPOS_0004382; BBPOS_0004383-BBPOS_0004384; BBPOS_0004383-BBPOS_0004384; BBPOS_0004382; BBPOS_0004422-BBPOS_0004423; BBPOS_0004622-BBPOS_0004627; BBPOS_0004628-BBPOS_0004648; BBPOS_0005113-BBPOS_0005114; BBPOS_0005115-BBPOS_0005116; BBPOS_0005113-BBPOS_0005114; BBPOS_0005112; BBPOS_0005115-BBPOS_0005116; BBPOS_0005112; BBPOS_0005121-BBPOS_0005122; BBPOS_0005123-BBPOS_0005137; BBPOS_0005123-BBPOS_0005137; BBPOS_0005121-BBPOS_0005122; BBPOS_0004628-BBPOS_0004648; BBPOS_0004622-BBPOS_0004627; BBPOS_1632236-BBPOS_1632239; BBPOS_1632240; BBPOS_1632240; BBPOS_1632236-BBPOS_1632239, *etc.*]

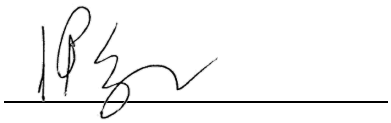
[10] [*See, e.g.,* BBPOS_0004382; BBPOS_0004383-BBPOS_0004384; BBPOS_0004399-BBPOS_0004406; BBPOS_0004390; BBPOS_0004413-BBPOS_0004417; BBPOS_0004391-BBPOS_0004397; BBPOS_0004388; BBPOS_0004407-BBPOS_0004410; BBPOS_0004419; BBPOS_0004389; BBPOS_0004398; BBPOS_0004418; BBPOS_0004412; BBPOS_0004411; BBPOS_0004385-BBPOS_0004387; BBPOS_0004422-BBPOS_0004423; BBPOS_0004622-BBPOS_0004627; BBPOS_0004628-BBPOS_0004648, *etc.*]

## 8 CONCLUSIONS

193. Within a reasonable degree of professional certainty, I believe that Ingenico has misappropriated these 5 trade secrets they received from BBPOS to produce their iTMP line of products.

194. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

Executed this 16th day of February 2022.



Ivan Zatkovich

# Exhibit A - CV of Ivan Zatkovich

eCompCONSULTANTS
INFORMATION TECHNOLOGY & PATENT LITIGATION
TELECOMMUNICATIONS ▪ INTERNET ▪ EFORENSICS

## *Ivan Zatkovich*

*301 W. Platt Street, #365 ■ Tampa, FL 33606 ■ (813) 601-8142 ■ ivanzat@ecompconsultants.com*

Ivan Zatkovich has 30 years experience in computer science, computer networks, and software engineering. Mr. Zatkovich is the principal consultant with eComp Consultants, a technology consulting firm specializing in intellectual property consulting for the telecommunications, web publishing, and eCommerce. Mr. Zatkovich has 11 years experience providing expert testimony for eCommerce disputes, eForensices, patent litigation, and gaming systems. Ivan has provided numerous expert reports, depositions, and trial testimony.

He is an industry speaker and consultant in eCommerce, Network Communications, Commercial website development, Search engine optimization, eBusiness practices and standards for corporations such as McGraw-Hill, Caldwell Banker, and Amazon.com. In the role of eBusiness Director and consultant he has designed and implemented many commercial websites and eCommerce businesses involving online coupons, promotion codes, aggregate buying, cross-sell / up-sell, retail purchasing practices, Merchant accounts, Fulfillment, for B2B and B2C business models.

| | | |
|---|---|---|
| **Director / Project Manager** | 13 years | Managed 6 to 35 member teams. Software development, systems Integration, critical path management, PMP/CMM methodologies. |
| **Telecommunications** | 11 years | Computer Telephony Integration (CTI) since 1984. Call Center, IVR, ACD, VoIP, and Unified Messaging (Verizon, Bell Canada, PTT Netherlands, and Mercury Communications). |
| **Internet, ecommerce** | 12 years | Commercial & Storefront websites, eMarketing and personalization, Search engine optimization, Online Catalog, B2B & B2C Portals, ERP and Supply chain, , Enterprise Search Engines, Certified Websphere Solutions Expert, IIS Server and app Server implementations. |
| **Social Media, Gaming, Media Distribution** | 7 years | Network Gaming, Social Media Platforms, Media Streaming, Web publishing, Video on Demand. Developed early High speed parallel and serial communication drivers, analog to digital signal processing for peripheral and device control. (Digital Equipment Corporation, Tanning Technology, Utility partners) |
| **Expert Witness** | 9 years | Expert testimony and damage assessment for ecommerce litigation, internet & software copyright, telecommunications. Over 17 cases providing expert reports, depositions, & trial testimony. |

**Education:**

| | | |
|---|---|---|
| B. S. Computer Science | 1980 | University of Pittsburgh |
| Minor in E.E. digital circuit design | | |
| Master's study and thesis in Computer Networks | 1981 | University of Pittsburgh |
| Results published in Byte Magazine | | |

**Summary of Case Experience:**

**Logiclink v. Keylink**

| | |
|---|---|
| Jurisdiction: | U.S. District Court, Southern California, Santa Anna |
| Client: | Logiclink |
| Nature of Case: | Ecommerce Patent Litigation |

| | |
|---|---|
| Nature of Engagement: | Testifying expert on the technology and business practices to provide Hotel business centers with connectivity and business applications for hotel guests. Patent Included capability for Hotel to configure their own customized advertising and banner ads. Infringement expert report and Trial testimony. |
| Represented by: | Peter Musielski, Esq. |

**eWinWin v. Groupon.com**

| | |
|---|---|
| Jurisdiction: | U.S. District Court, Middle district of Florida |
| Client: | eWinWin |
| Nature of Case: | eCommerce Patent Litigation, |
| Nature of Engagement: | Testifying expert for eCommerce practices, online coupons, aggregate buying, B2B and B2C transactions., |
| Represented by: | Colby Springer, Esq. |

**PointServe v. Scient**

| | |
|---|---|
| Jurisdiction: | American Arbitration Association – Houston, Texas |
| Client: | Scient (defendant - software vendor) |
| Nature of Case: | eCommerce Software Contract Dispute |
| Nature of Engagement: | Expertise for eCommerce business referral website. Evaluate web product development and standard business features provided by vendor, online directory for commercial services. Review plaintiff claims, failure to meet software specifications. |
| Represented by: | Steefel Levitt |

**Nuance Communications v. TellMe Networks (Microsoft)**

| | |
|---|---|
| Jurisdiction: | U.S. District Court, Delaware |
| Client: | TellMe Networks (Microsoft) |
| Nature of Case: | Patent Litigation |
| Nature of Engagement: | Provide expertise in speech recognition and call center technology, prior art investigation, opinions on patent validity and non-infringement. |
| Represented by: | Patterson Belknap Webb & Tyler, LLP |

**Katz v. Fifth Third Bank (and 5 other defendants)**

| | |
|---|---|
| Jurisdiction: | U.S. District Court, Northern California |
| Client: | Defendants |
| Nature of Case: | Patent Litigation for Call Center, and Telecom technology |
| Engagement: | Expertise in Call center systems for Banking, Mortgage, credit cards. Analyze client call center applications and prepare non-infringement and rebuttal expert report. |
| Status: | Open |
| Represented by: | Vorys LLP, Wood Herron and Evans. |

**Professional Experience**

**2013 – 2016** **Fund Leader** ***New York, NY***

***Vaud Advisors* - Social Media Technology Fund**

Fund Leader and technology advisor.

Define criteria for selecting and evaluating Social Media technology start ups including technologies for Gamifying, Charitable Contributions, Entertainment, and Mobile Applications. Provided specific evaluation of 'SWOT' characteristics and Business models (e.g. branding, freemiums, partner integration)

**2000 - Present** **Principal Consultant**
***eComp Consultants*** ***Tampa, FL***

Provide consulting on design and development of software product development, and ecommerce technology. Provide technology consulting and expert support for telecommunications, internet, and ecommerce applications in the areas of:

- Patent portfolio evaluation, market valuation, identification of potential prior art, and licensing opportunities.
- Independent consulting for Alcatel-Lucent (AT&T Bell Labs) to review, recommend, and rate Alcatel's extensive Patent portfolio specifically in the area of telecommunications and web-based ecommerce.
- Patent litigation consulting for telecommunications and internet technology patents.
- Software contract and due diligence consulting for disputes involving custom software projects.
- Internet publishing and copyright consulting for web content management and licensing.
- Commercial websites, search engine, and web marketing strategies.

Sample clients include: Alcatel-Lucent, LogicLink, NWI Corp, Jones Day, Finnegan Henderson, Fish & Richardson, Cozen O'Connors, Morrison Foerster

**2007- 2008** **Global Web Markets Manager**
***Smith & Nephew Inc.*** ***Clearwater, FL***
Smith & Nephew is a global manufacturer of medical, wound, orthopedic, and surgical supplies. Smith & Nephew has facilities in 33 countries and maintains 35 websites for different product areas and country marketing.

- Design the S&N Corporate web architecture for all intranet, extranet, and Internet sites.
- Direct a staff of web professionals in developing and managing 35 individual S&N web sites maintained in the US and 14 countries for both S&N clients and internal users.
- Manage and maintain the 650 domains currently owned by Smith & Nephew
- Optimize search engine ranking and website meta tagging.
- Develop corporate technology and IT standards and policies including:
  - Create Web hosting agreements and IT service contracts (SLAs)
  - Define Web development standards
  - Research page ranking, redirection, and domain optimization strategies.

**2002 – 2007** **eBusiness Director**
***Eva-Tone Inc*** ***Clearwater, FL***
Eva-Tone is a $45M commercial multimedia company specializing in ecommerce and marketing solutions, media and content customization, order processing and fulfillment, print & web publishing, CD & DVD manufacturing.

- Set strategic direction of client service offerings in ecommerce and Content Management.
- Designed and Implement High Volume web publishing system.
- Projects proposed and delivered include:
  - McGraw-Hill – Enterprise Content Management and automated Web publishing system.

- ProMarine USA – B2B storefront, complex part search, order processing and fulfillment.
- Resource Rabbit – Sales collateral manager. This system retrieves user preferences and client profile information to generate customized electronic presentations and brochures.
- AAA Travel – CRM sales collateral system, instant customized brochures, print-on-demand.

**1999 to 2002** **eBusiness Engagement Manager**
***Tanning Technology and IMRGlobal***
Tanning Technology is an international systems integrator specializing in network infrastructure and channel integration for financial markets and the insurance industry. Proposed and managed eBusiness and Internet infrastructure projects focused on client and user personalization, data mining, and web marketing practices.

Client web engagements proposed and managed:

- Eckerd Pharmacy – Developed eBusiness and web marketing for personalized client web content.
- GEICO – Developed a personalized policyholder web service that uses clients' account history and demographics to customize product literature for cross-sell & up-sell opportunities.
- Hartford Insurance – Integrated 3 tier online claims processing & subrogation with legacy systems
- Citicorp Bank – Designed mortgage and loan payment system; managed sale of loan contracts
- Smith Barney – Developed remote access network infrastructure & wireless PDA financial system
- Medwerks – Created medical insurance clearinghouse, MDs centrally process insurance payments

**1996 –1999** **Director of Network & Customer Support**
***Utility Partners, Inc.*** ***Tampa, FL***
Utility Partners developed and customized utility company software for scheduling service appointments, assigning and routing mobile workforce (field technicians), and dispatching and coordinating resources during power outages.

- Managed client engagement leads for development of client customizations and standardization of product features.
- Setup client support department to manage UP's growing client and product base. This department:
  - Provided Network and IT management for over 800 customer users
  - Created a 24x7 Helpdesk for Utility dispatch centers and IT departments
  - Developed system and network monitors for real-time client system monitoring
  - Received trouble calls & provided live client support for the following applications:
    - MobileUP – Mobile Workforce Dispatcher
    - TroubleUP – Power outage manager
    - Smartnom – Web Gas Auction System for private gas auctions
    - CAS – Customer Appointment and Call Center application

**1987-1996** **Project Manager: Telecom Software**
***GTE Data Services (currently Verizon)*** ***Tampa, FL***
GTE was an $8B Telecommunications company and formed the Commercial Services

division to customize their telephone business software to sell to other telephone companies internationally.

- Managed large product implementations for Telecom billing, service order, and Switch management.
- Developed Service manager, an early videotext hypermedia application for online ordering of Telecom equipment and services.
- Specific projects managed:
  - Mercury Communication – setup Service Bureau to bill their new intelligent network services.
  - PTT Netherlands – managed $5M inter-carrier billing project for PTT Netherlands
  - Bell Canada – developed Pricing Plan & Table Management system
  - Deutsche Telekom – designed and developed a German billing prototype system.
  - Received GTE personal best award.

**1980 – 1987** **Software Engineer / Technical Lead**

***Digital Equipment Corp.*** ***Maynard, MA***

As Technical Lead, led the development of manufacturing automation and control software. Projects Included:

- Supply Chain, CAD/CAM – Designed system to retrieve parts information from circuit board CAD design. Developed a parts ordering system based on expected volumes.
- Pick & place automation – Automatically program IC insertion machines. Develop system to determine IC insertion order, generate machine instructions, and download program to machine.

As Software Engineer for embedded controllers and video subsystems, designed and developed PDP-11 operating system and device drivers.

- Enhanced high-speed DecNet drivers for Synch, Asynch, and Parallel communications.
- Designed/developed disk firmware and drivers for floppy & Winchester disks.
- Developed Hypermedia and Windows Manager for PRO-350 microcomputer.
- Developed and enhanced video controller firmware and layered graphics system for PRO-350.
- Developed Computer Telephony Integration (CTI) product for telephone answering machines.

**Technical Tools & Environments**

- Unix, Windows, NT, Microsoft .NET, C#, C/C++, VB, Assembler
- Java, Java script, JSP, IBM Websphere, Microsoft Commerce engine, Blue Martini
- SQL, Oracle, Foxpro, DB2, Crystal Reports, Active-Reports, MQ Series, Tuxedo, Documentum, Interwoven

**Certifications & Publications**

- Industry Speaker: Internet Publishing standards (Momentum conference)
- Department of Justice: Proposal for internet forensics technology
- IBM Websphere – Certified ecommerce Solutions Expert
- Byte Magazine – Published Network Design articles
- Sync Magazine – Published Programming Techniques and Tutorials
- IEEE SigGraph – Presented ICGS Computer Graphic Standards for IEEE SigGraph conference.

- CMM & PMP – Project management methodologies
- ISO and ANSI – On International ISO and National ANSI Committees for Disk and Media format standards

## *Ivan Zatkovich - Consulting, Testimony & Publications*

**Non-litigation consulting work:**

The following lists the companies for whom I was employed or for whom have consulted for non-litigation matters for at least the last 5 years.

- eComp Consultants – 2001 - present
- Alcatel – Lucent – 2008 - 2011
- Verisign - 2011
- Smith & Nephew – 2001 - 2002
- Evatone, Inc – 2002 - 2007
- McGraw Hill - 2002 - 2006
- AAA Travel – 2004 - 2005
- Pro Marine – 2005 - 2006
- Wachovia - 2007
- Capitol One – 2007
- Network 1 Security Solutions – 2012 – 2013 – evaluation of call center technologies
- LGreen Deals – 2012 – 2015 – evaluation of technologies and startup companies for investment opportunities
- Vaud Advisors Social Media Technology Funds – 2014 – 2016 – identification and selection of technology startup companies
- 3M Futures South Africa – 2014 – 2016 – Mobile payment authentication
- Logiclink – 2015 – IP evaluation for hotel business kiosks
- Mario Constanze – 2019 – Evaluation of Blockchain and Cryptocurrency technology

**Certifications & Publications:**

- Industry Speaker: Internet Publishing standards (Momentum conference) – 2006
  Publishing high volume content for print and web
- Department of Justice: Proposal for internet forensics technology – 1999
  Methods for fingerprinting and detecting copyright material during transmission
- IBM Websphere – Certified ecommerce Solutions Expert - 2000
- Byte Magazine – Published Network Design articles. 1980
- Sync Magazine – Published Programming Techniques and Tutorials 1980-1982
- IEEE SigGraph – Presented ICGS Computer Graphic Standards for IEEE SigGraph conference. 1984
- CMM & PMP – Project management methodologies – 1996 - 1998
- ISO and ANSI – On International ISO and National ANSI Committees for Disk and Media format standards – 1981-1984

**Court Testimony Experience:**

- *Logiclink v. Keylink*, CV07-1056-DOC(MLGx) (C.D. Cal.)
  Patent infringement dispute for the Plaintiff, concerning eCommerce for Kiosks for Hotel Business centers. Bench trial: The judge ruled in favor of and awarded damages to the Plaintiff.
  January 2009-February 2009

- *3M Future Africa (PTY) LTD v. The Standard Bank of South Africa LTD, MTN Group Limited and MTN Mobile Money SA (PTY) LTD,* Case #: Patent 2002/2337 (South African High Court) Testifying expert patent infringement and patent validity for the Plaintiff for Online Payments Technology. Bench trial: Testified on the stand for 2.5 days (bench trial). The judge ruled in favor of the Plaintiff on all points of contention for both Infringement and Patent Validity and cited the expert's opinion in each ruling.
  March 2011 -July 2012

- *Kenneth Nkosano Makate v Vodocom*, Case number: 08/20980 (SOUTH GAUTENG HIGH COURT, JOHANNESBURG, SOUTH AFRICA)
  Testifying Expert with Report and Trial Testimony opining on the viability of Makate's concept idea for the "Buzzing Option" that became the "Please Call Me" product; explain other similar concepts used by other service providers and how other business would reward for such a concept that significantly increase revenue and profit for the company. Technology in this case was Unstructured Supplementary Service Data (USSD) a protocol used by GSM cellular telephones to communicate with service provider's computers. USSD can be used for WAP browsing, prepaid callback service, mobile-money services, location-based content services, menu-based information services, and as part of configuring the phone on the network. The judge ruled that Makate should be granted an arbitration hearing to receive compensation for his invention.
  May 2013 – August 2013

- *Black Hills Media, LLC v. Samsung, et al.* INV# 337-TA-882 *[ITC case]*
  *Samsung, LG, Panasonic, Toshiba, Sharp*
  Testifying Expert on infringement and patent validity for the plaintiff, regarding media streaming, playlists, and geolocation technology. Testified at the ITC hearing. The ITC administrative judges ruled that the accused products did meet the limitation of the claims, but where not in an infringing state at the time of importation.
  July 2014-October 2016

- *CertusView v. S&N*, 2:13-cv-346 (MSD) (LRL) (Virginia)
  Testifying Expert regarding geolocation. U.S. PATENT # 8,265,344, 8,290,204, 8,340,359, 8,407,001, AND 8,532,341. Testifying Expert in inequitable conduct trial concerning impact of undisclosed prior art.
  March 2014-March 2016

**Case History:**

The following is a list of matters in which I have testified and/or provided expert services (the underlined party identifies which party I was retained by), including all such matters where I have provided written or oral testimony in the past five years.

• *Zamora v. CBS Radio et.al.* 09-20940-CIV-MORENO (S.D. FL 2010) (settled 2010)
*Last.fm, Ltd., CBS Radio, Inc., CBS Corp., Slacker, Inc., Pandora Media, Inc., Rhapsody America LLC, Realnetworks, Inc., DKCM, Inc., AOL, LLC, Accuradio, LLC, Yahoo! Inc. and Soundpedia, Inc.*
Expert for Plaintiff on Patent Infringement for Internet Radio Technology. Provided expert report and deposition regarding the use of streaming media and Web Radio players.
*January 2010*-March 2010

• *Nuance v. Tellme (a Microsoft subsidiary)* C.A. No. 06-105-slr (S.D. Del 2009) (settled 2010)
Expert for Defendant on Speech Recognition software. Provided expert report and deposition on non-infringement and invalidity for speech recognition used in automated phone directories.
May 2009 – August 2010

• *Ronald A. Katz v. DHL Express*, 2:2007 CV 02192 (N.D. Cal.)
Patent infringement dispute concerning call processing center and call routing patents. Developed non-infringement reports and deposed.
March 2012 – January 2012

• *Ronald A. Katz v. Cox Communications*, 2:2007 CV 02299 (N.D. Cal.)
Patent infringement dispute concerning call processing center and call routing patents. Developed non-infringement reports and deposed.

• *Ronald A. Katz v. Earthlink*, 2:2007 CV 02325 (N.D. Cal.). Patent infringement dispute concerning call processing center and call routing patents. Developed non-infringement reports and deposed.
July 2012 – August 2012

• *Ronald A. Katz v. Fifth Third Bank*, 2:2007 CV 04960 (N.D. Cal.)
Patent infringement dispute concerning call processing center and call routing patents. Developed non-infringement reports and deposed.
January 2009-August 2010

• *Ronald A. Katz v. Huntington National Bank*, 2:2007 CV 04960 (N.D. Cal.)
Patent infringement dispute concerning call processing center and call routing patents. Developed non-infringement reports and deposed.
January 2009 – November 2009

• *Ronald A. Katz v. Echostar*, 2:2007 CV 06222 (N.D. Cal.) (pending)
Patent infringement dispute concerning call processing center and call routing patents. Developed non-infringement reports and deposed.
January 2009 – July 2012

• *Logiclink v. Keylink*, CV07-1056-DOC(MLGx) (C.D. Cal.)
Patent infringement dispute concerning eCommerce for Kiosks for Hotel Business centers. Developed infringement report and invalidity rebuttal report. Trial Testimony.
January 2009 – February 2009

• *eBay v. IDT corp,* IDT 4:08-cv-4015-HFB (W.D. Ark) (settled 2010)
Patent infringement. Perform analysis of Voice Over IP providers and prior art candidates; provide non-infringement (suit) and infringement (counter suit) expert reports.
March 2010 – June 2010

• *i2 Technologies, Inc. v. Oracle Corporation,* 6:09-CV-194-LED (E.D. Tex)
Patent infringement, testifying expert for Supply Chain Management software, Manufacturing Automation, and Sales projection system software. Provide infringement and Invalidity rebuttal.
May 2010 – March 2011

• *IslandIP Inc. v. Deutsche Bank Corporation,* 1:09-cv-04673-VM (S.D. NY)
Patent infringement, testifying expert for Financial Transaction, Omnibus account consolidation and settlement for Banks and Broker Dealers. Provide infringement and Invalidity rebuttal reports.
September 2009 – February 2012

• *3M Future Africa (PTY) LTD v. The Standard Bank of South Africa LTD, MTN Group Limited and MTN Mobile Money SA (PTY) LTD,* Case #: Patent 2002/2337 (South African High Court)

Patent infringement, testifying expert for Online Payments Security. Provide infringement, validity and trial support.
March 2011-July 2012

• *Cohen v. US,* 07-154-C (*U.S. District Court of Federal Claims*)
Copyright infringement testifying expert for Cohen. US Government FEMA distributed copy righted documents without Cohen's knowledge or approval. Provide expert report, damages and deposition.
June 2009 – February 2011

• *SFA v. 1-800-Flowers, et.al,* 6:2009-cv-00340-LED (E.D. Tex)
• *SFA v. Barnes & Noble, et.al,* 6:2011-cv-00399-LED (E.D. Tex)
*1-800-Flowers.com, Inc.,The Plow & Hearth, Inc., including D/B/A Wind & Weather, Inc.*
*The Popcorn Factory, Inc., Winetasting Network, Inc., The Children's Group, Inc.,*
*Problem Solvers, Inc., Barnes & Noble, Inc., Barnesandnoble.com LLC, Blockbuster, Inc.,*
*BUILD-A-BEAR WORKSHOP, INC., CDW Corporation*
*GameStop, Corp., GameStop, Inc., GameStop.com, Inc., Gander Mountain Company,*
*Overton's, Inc., J & R Electronics, Inc., Newegg, Inc., Newegg.com, Inc.*
*Northern Tool & Equipment Company, Inc., Northern Tool and Equipment Catalog Co.*
*Office Depot, Inc., Omaha Steaks International, Inc., OmahaSteaks.com, Inc.*
*The Timberland Company, Tupperware Brands Corporation, Tupperware.com, Inc.*
Patent infringement, for Computerized Sales Force Automation System. Testifying expert for joint invalidity (9 defendants) and non-infringement (5 defendants).
May 2011 – October 2011

• *Bloom v. Intuative*, 06 CV 6301 (S.D. NY)
Software contract dispute, testifying expert concerning Document Management and Publishing for Pharmaceuticals and Regulatory agencies. Comparison of product features and software reuse.
January 2009 – April 2010

• *Dallal v. New York Times,* 1:2003-cv-10065 (S.D. NY) (settled 2008)
Testifying Expert, Deposition, for Web Copyright. Provided expertise in standard media publishing practices, content licensing, and web content customization.
January 2009 – April 2010

• *Bear Creek Technologies v. Verizon Services Corp.,* 1:11-cv-00880-TSE -JFA (E.D. Virginia)
Testifying Expert, provided infringement and invalidity of VoIP related patents.
October 2011 – March 2012

• *Cequint v. Apple,* 1:2011-cv-01224 (Wilmington) (pending)
Testifying Expert. Performed code review of Cequint's City ID Software product to determine if and how they are practicing the key limitations of the patents-in-suit focused on Caller ID, CND Messages, City, State lookups, WAP interfaces, Database synchronicity (automatic update of database), Display of incoming caller information and Call Answer and set-up code.

July 2012 – August 2012

• *Tel-tron Corporation v. Stanley Security Solutions d/b/a Stanley Healthcare Solutions,* 6:2011-cv-01448 (M.D. Florida)
Testifying Expert regarding network systems for displaying data relating to emergency call systems.
August 2012

• *Microsoft v 5R Processors*, LTD. *and Thomas Drake*, 3:12-cv-00263-slc, (W.D. Wisconsin)
Testifying Expert. Evaluate the Microsoft Windows Professional XP OEM Licensing Agreement and Microsoft Refurbished PC Licensing Guidelines and opine on 5R adherence.
July 2012 – August 2012

• *Walker Digital, LLC v. Fandango, Inc., et al.,*1:11-cv-00313-SLR (Delaware)
*Facebook Inc., Fandango, Inc., Expedia, Inc., Amazon.com, Inc., American Airlines, Inc., eBay Inc., and Zappos.com, Inc.*
Testifying Expert regarding e-commerce, shopping carts, e-marketing, and the promotion of financial products such as credit and debit cards for retail, business-to-business ("B2B"), and financial industries. Provided Infringement (Amazon, Expedia, Zappos) Reports and Invalidity Rebuttal Report.
December 2011 - June 2013

• *Trialcard v. P.S.K.W. & Associates,* 3:11-cv-05693-FLW-TJB (NewJersey)
Testifying Expert for patent re-exam concerning eCommerce technology based on a method of dispensing, tracking, and managing pharmaceutical products by communicatively linking prescribers and pharmacies to a central computing system
August 2012 – October 2012

• *Catalina Marketing Corporation v. Coupons, Inc.*, JAMS Reference No. 11000654507 (San Francisco)
Testifying Expert to analyze the following Coupons systems, Microsite System & Brandcaster™ System and determine if either system is an "Infringing Capable System". Provided Expert Report and Mediation Trial support.
September 2012 – November 2012

• *Soverain v. Euromarket Design*, et al. 6:12-cv-00145-LED (E.D. Texas) (pending)
Testifying Expert regarding e-commerce systems/online shopping carts technology.
December 2012 – January 2013

• *e-LYNXX Corporation v. InnerWorkings, Inc., et al.* 1:10-CV-2535 (M.D. PA)
*InnerWorkings, Standard Register, Cirqit.com and R.R. Donnelley & Sons Co.*
Testifying Expert regarding supply chain, custom print jobs, matching job requirements to vendors technology. Provided Infringement, Invalidity and Non-Infringement Rebuttal Reports and deposition.
March 2012 – May 2013

• *Joao Bock Transaction Services v. USAmeribank, et al,.* 8:11-cv-00887-MSS-TGW (Tampa)
*USAmeribank, Everbank and BankFirst*
Testifying Expert regarding secure transactions on electronic financial accounts.
March 2013 - May 2013

• *Lodsys v. Brother Intl Group et al.,* 2:11-cv-90(JRG) (E.D. Texas)
*Brother Intl Corp, Canon U.S.A., Inc., Hewlett-Packard Company, Hulu, LLC, Lenovo (United States) Inc. Lexmark International, Inc., Motorola Mobility, Inc., Novell, Inc., Samsung Electronics Co., LTD., Samsung Electronics America, Inc., Samsung Telecommunications America, LLC, Trend Micro Incorporated*
Testifying Expert regarding interactive media enabling CRM technology. Provided Invalidity Rebuttal Reports and deposition testimony
April 2013 – October 2013.

• *Progressive v. State Farm, et al.* CBM2012-00003, CBM2013-00004
*State Farm Insurance, Hartford Insurance, Liberty Mutual Insurance, Safeco Insurance, Ohio Casualty Insurance, Open Seas Solutions, Octo Telematics*

Testifying Expert regarding e-Commerce and vehicle telematics systems. Provided Patent Re-exam Declarations.
March 2013 – February 2014

• *Black Hills Media, LLC v. Samsung, et al.* INV# 337-TA-882 *[ITC case]*
*Samsung, LG, Panasonic, Toshiba, Sharp*
Testifying Expert regarding Google Latitude running on smartphones and smart phones running software which can direct the streaming of content between, for example, a laptop and a TV or Blu-ray player. Providing device analysis and code review, Infringement Reports and Rebuttal Reports.
July 2014 – October 2016

• *Rembrandt Social Media, LP v. Facebook, Inc. and ADDTHIS, Inc.,* 1:13cv158 (TSE/TRJ) (Virginia)
Testifying Expert regarding social media user content and 3rd party interaction -the diary page and "Like" feature. Provided Non-Infringement Report and deposition.
May 2013 – June 2014

• *Indacon, Inc v. Facebook, Inc.* 5:130cv966-OLG (TX)
Testifying Expert regarding automated creation of user-defined hyperlinks in a database. Provided Non-Infringement Report and deposition.
March 2014 – August 2014

• *GeoTag, Inc v. Frontier Communications Corp. et al.* 2:10-cv-00265 *(TX) [Multiple cases]*
*Macy's, Genesco, AGCO Corp., The Rockport Company, LLC., Hennes & Mauritz LP (H&M), Gander Mountain Company, Godiva Chocolatier, DSW, National Interlock Systems*
Testifying Expert regarding e-Commerce, Geo location, store locator-based technology. Provided Non-Infringement Reports for multiple defendants.
July 2013 – January 2014

• *CareerFairs.com v. United Business Media, LLC, et al.,* 11-20329-CIV-KING (S.D. Florida)
Testifying Expert regarding videoconferencing, virtual career fairs technology.
June 2012

• *Transauction LLC v. eBay Patent 7343339* 3:2009cv03705 (N.D. CA)
Expert regarding computerized electronic auction payment systems.
May 2010 – June 2010

• *Ameranth v. Six Continents Hotels,* 09-CV3819 (GA)
Testifying Expert regarding Internet Advertising & Website Technology. Expert Report and deposition.
March 2010 – June 2010

• *Carolina Power & Light Company v. Aspect Software* 5:08-cv-00449-BO (E.D. NC)
Testifying Expert in telecommunications technology.
October 2009 – December 2009

• *Dominion v. Aspect Software,* 3:08-cv-00737-REP (Virginia)
Testifying Expert in telecommunications and IVR call processing. Provided expert declaration and testimony.
June 2009

• *ComplementSoft v. SAS Institute,* 12-C-7372 (N.D. IL)
Expert regarding software development tools, database manipulation, SAS programming.
March 2013-February 2015

• *Moricz v. Google,* 10-cv-01240-RSL (Seattle)
Expert regarding use of deep search methodology. Claim Construction support.
August 2011 – September 2011

• *Media Rights Technologies v. Capital One, et al.,* 1:13cv476-AJT/TRJ (Virginia)
Expert regarding secure internet Banking System technology USPatent# 7316033.
August 2013 – January 2014

• *CertusView v. S&N,* 2:13-cv-346 (MSD) (LRL) (Virginia)
Testifying Expert regarding geolocation. U.S. PATENT # 8,265,344, 8,290,204, 8,340,359, 8,407,001, AND 8,532,341. MARCH 2014-DECEMBER 2014

• *Bright Edge v SearchMetrics,* 3:14cv-01009 (San Francisco)
Testifying Expert regarding search optimization, USPatent# 8478700, USPatent# 8478746.
July 2014 – November 2014

• *Black Hills Media, LLC v. Sonos,* 2:14-CV-00486 SJO (PJWx) (California Western Division)
Testifying Expert regarding delivery of stored and streaming audio content over wide-area and local area networks. U.S. PATENT # 6,757,517, 7,236,739, 7,742,740 and 6,826,283
July 2014 – October 2016

• *Yamaha Corp of America v. Black Hills Media, LLC,* IPR2013-00593, IPR2013-00594, (PTO)
Testifying Expert regarding software which can direct the streaming of content between devices, for example, a laptop and a TV or Blu-ray player. Providing Inter Partes Review Declaration Reports.
U.S. PATENT # 8,050,652, 8,045,952
July 2014 – August 2014

• *Samsung Electronics, et al. v. Black Hills Media, LLC,* IPR2014-00737, IPR2015-00334, IPR2014-00740, IPR2015-00340, IPR2014-00735 (PTO)
Testifying Expert regarding software which can direct the streaming of content between devices, for example, a laptop and a TV or Blu-ray player. Providing Inter Parties Review Declaration Reports.
U.S. PATENT # 8,050,652, 8,045,952, 6,618,593
July 2014-October 2016

• *GT Nexus v Inttra, Inc.*, 4:11-cv-02145-SBA (California Northern Division)
Testifying Expert regarding networks & systems used for logistics management. Provided Declaration for CBM petitions. U.S. Patent No. 7,761,387, 7,752,142, 7,827,119, 7,756,794
January 2015 – Sept 2016

• *MyKey Technology, Inc. v. Intelligent Computer Solutions, Inc.,* 2:13-ml-02461-AG (PLAx) MDL NO. 2461 (Central California Southern Division)
Testifying Expert regarding hard drives and other mass storage devices, how to access "hidden" storage and cleaning a disk. Provided Validity Rebuttal and MSJ Declaration. U.S. Patent No. 6,813,682, 7,228,379
March 2015 – June 2015

• *Biosignia Inc. v. Life Line Systems*, 1:12-cv-01129-UA-JEP (MDL NC)
Testifying Expert regarding Contract Dispute concerning Health Risk Assessment System.
July 2015 – January 2016

• *Kenneth Nkosano Makate v Vodocom*, Case number: 08/20980 (SOUTH GAUTENG HIGH COURT, JOHANNESBURG, SOUTH AFRICA)
Testifying Expert with Report and Trial Testimony opining on the viability of Makate's concept idea for the "Buzzing Option" that became the "Please Call Me" product; explain other similar concepts used by other service providers and how other business would reward for such a concept that significantly increase revenue and profit for the company. Technology in this case was Unstructured Supplementary Service Data (USSD) a protocol used by GSM cellular telephones to communicate with service provider's computers. USSD can be used for WAP browsing, prepaid callback service, mobile-money services, location-based content services, menu-based information services, and as part of configuring the phone on the network.
May 2013 – August 2013

• *Parus Holdings, Inc. v. Airtran Airways, Inc., et al.,* 8:11-cv-01685 (MDL FL)
Testifying Expert regarding expertise in voice recognition, isolated word recognition, natural speech recognition, unified messaging systems, interactive voice response systems, and travel reservation systems for invalidity analysis report. Provided invalidity report.
March 2012 – June 2012

• *Versata Software, Inc. et al. v. Callidus Software Inc.*, C.A. No. 1:10-cv-00781-SLR (Delaware)
• *Versata Software, Inc. et al. v. Callidus Software Inc.*, C.A. No. 1:12-cv-00931-SLR (Delaware)
Testifying Expert regarding expertise in sales force automation, workflow analysis, process automation, event management and detection, rules-based systems. Provided analysis for infringement report.
August 2014 – November 2014

• *BuySafe v. Google*, 3:13-cv-00781-HEH (Richmond, VA)
Testifying Expert regarding patent litigation concerning eCommerce specifically to determining the efficacy (effectiveness, strength, success, etc.) of offers that are related to online transactions. The efficacy is determined based on a user's actions in the online environment.
May 2014 – September 2014

• *Phoenix v. General Motors*, 2:13-cv-1093 (ED Texas)
Testifying Expert regarding patent litigation concerning US Patent 5,987,434 - Apparatus and method for transacting marketing and sales of financial products, US Patent 7,890,366 - Personalized communication documents, system and method for preparing same, US Patent 8,352,317 - System for facilitating production of variable offer communications.
May 2015 – August 2015

• *Macropoint v. FourKites*, 1:15-cv-01002 (ND Ohio)
Testifying Expert regarding patent litigation concerning U.S. Pat. Nos. 8,604,943, 9,070,295, 9,082,097, 9,082,098, and 9,087,313. At a very high level, the patents are directed to methods and systems for monitoring and tracking the location of a vehicle or freight in transit. Evaluated the claims of the asserted patents to opine on whether the claims are: 1) directed to an abstract idea, and 2) in the event the court would find the claims directed to an abstract idea, how the limitations of the claims transform that abstract idea into patentable subject matter based on the Supreme Court Alice decision being barely a year old.
August 2015 – September 2015

• *Macropoint, LLC, v. Ruiz Food Products, Inc.,* 6:13-cv-1133-RSW-KNM (E.D. of Texas Tyler Division)
Testifying Expert regarding patent litigation.
October 2017 – May 2018

• *Longitude v. Apple*, 3:14-cv-04275 (San Francisco)
Testifying Expert regarding patent litigation concerning Flash Memory Technology. Provided claim construction support.
April 2015 – February 2016

• *Global Tel Link Corp. v. Securus Technologies, Inc.*, CBM2015-00145 (US PTAB)
Testifying Expert regarding a CBM concerning the 7,860,222-patent titled "Systems and Methods for Acquiring, Accessing, and Analyzing Investigative Information," generally directed to acquiring, accessing, and analyzing investigative information or phone call monitoring using speech recognition/transcription of inmates within a jail.
October 2015 - December 2015

• *Global Tel Link Corp. v. Securus Technologies, Inc.*, IPR2015-01222 (US PTAB)
Testifying Expert regarding an IRP concerning the patent 8,750,486 titled "Call Center for offering goods and services to an inmate population," generally directed to acquiring, accessing, and analyzing investigative information or phone call monitoring using speech recognition/transcription of inmates within a jail.
January 2016 – May 2016

• *Global Tel Link Corp. v. Securus Technologies, Inc.*, IPR2015-01225 (US PTAB)
Testifying Expert regarding an IRP concerning the patent 8,886,663 titled "Multi-party Conversational analyzer and logger," generally directed to acquiring, accessing, and analyzing investigative information or phone call monitoring using speech recognition/transcription of inmates within a jail.
January 2016 – June 2016

• *CaptionCall v Ultratec, Inc*, IPR2015-01355 (5,974,116), IPR2015-01357 (6,934,366), IPR2015-01358 (7,006,604), IPR2015-01359 (6,493,426), IPR2015-00636 (8,917,822), IPR2015-00637 (8,908,838) (US PTAB)
Testifying Expert regarding IPRs concerning call routing/switching that happens when a deaf or hearing-impaired individual makes a phone call.
October 2015 -March 2017

• *Centrak, Inc. v Sonitor Technologies*, Inc., 14-183-RGA (Delaware)
Testifying Expert regarding patent litigation concerning an application on a smart phone which allows a person to walk into a hospital, have infrastructure pick up patient information and send it to say, an MRI office, so office is ready for the patient when patient arrives. Expertise in low energy blue tooth and way finding and Ultrasonic Tags communicate with Stations. Provided infringement report and invalidity rebuttal.
February 2016 – May 2018

• *Intellectual Ventures II LLC v. T-Mobile USA, Inc., et al.,* C.A. No. 13-1633-LPS (D. Del.)*;*
*Intellectual Ventures II LLC v. Nextel Operations, et al., (on behalf of Sprint)* C.A. No. 13-1635-LPS (D. Del.)*;*
Testifying Expert regarding non-infringement of US Patent No. 6,115,737 and the 5,339,352 concerning accessing customer contact services over a network and a Directory assistance call completion via mobile systems.
March 2016 – October 2016

• *Askeladden LLC v. N5 Technologies LLC* IPR2017 – 7,197,297*;*
Testifying Expert regarding IPR of US Patent No. 7,197,297 concerning mobile messaging and authentication of mobile devuce users.

September 2016 – January 2018

• *Askeladden LLC v. Verify Smart Corp.* IPR2017 – 8,285,648*;*
Testifying Expert regarding IPR of US Patent No. 8,285,648 concerning electronic transactions and authentication of customers using mobile devices.
November 2016 – February 2018

• *Askeladden LLC v. Digital Verification*
Testifying Expert regarding patent litigation.
December 2017 – March 2018

• *Infinity Computer Products, Inc. v. Toshiba America Business Solutions, Inc.,* 2:12-cv-06796-NIQA (E.D. of Pennsylvania)
Testifying Expert.
November 2017 – May 2018

• *IXI Mobile (R&D) LTD. and IXI IP, LLC v. Blackberry Limited, et al.,* 2:15-cv-01883-JRG-RSP (E.D. of Texas Marshall Division)
Testifying Expert.
July 2016 – October 2016

• *Financial Information Technology v. Mark Lopez,* 8-15-cv-02784-T-30AEP (D.C. of Middle Florida)
Testifying Expert.
November 2016 – May 2017

• *Financial Information Technology v. iControl Systems,* 8:2015-cv-00190 (D.C. of Middle Florida)
Testifying Expert.
January 2018 -- current

• *Alexsam, Inc. v. Green Dot Corporation, Next Estate Communications, Inc., and Does 1 through 10, inclusive, Case No 2:15-cv-05742 CAS(PLAx) (C.D. of California Western Division)*
Testifying Expert.
January 2017 – July 2017

• *Wex Health, Inc., v. Alexsam, Inc.,* Case No 2:17-cv-00733-RWS-RSP (E.D. of Texas Marshall Division)
Testifying Expert.
May 2018 – January 2019

• *Bank of the West v. Trisport.com L.L.C., and Seton Claggett and Deborah Claggett,* Case No C20131286 (Superior Court of Arizona)
Testifying Expert.
November 2016 – May 2017

• *Noble Systems Corp. v. Kabbage, Inc.*
Testifying Expert.
November 2016 – February 2017

• *ComplementSoft LLC v. SAS Institute Inc.,* 1:2012-cv-07372 (D.C. of Northern Illinois)
Testifying Expert.
April 2013 – February 2015

• *Intellectual Ventures II LLC v. T-Mobile USA, Inc., et al, C.A No 13-1635-LPS; Intellectual Ventures II LLC v. Nextel Operations, et al, C.A No 13-1635-LPS; Intellectual Ventures II LLC v. United States Cellular Corp., C.A No 13-1637-LPS (District of Delaware)*
Testifying Expert.
March 2016 – November 2016

• *ConnectSoft, Inc. v. NEEO Inc.,* Case No. 2:16-cv-00548-JRG (E.D. of Texas Marshall Division)
Testifying Expert.
January 2017

• *LiverPerson, Inc. v. 24/7 Customer, Inc., Case No. 3:17-cv-01268-JST (*Northern District Court of California*); 1:14-cv-01559-RWS (*Southern District of New York)
Testifying Expert.
March 2018 - Current

• *Google LLC v. Performance Price Holdings, LLC and its affiliates (PPH),* 1:15-cv-09712-LGS (New York Southern District Court) – 0:2018bcaag01189; 0:2018bcaag01191 (U.S. Court of Appeals, Federal Circuit - CBM2016-00050 for Patent No. 8,799,059, CBM2016-00049for
Patent No. 7,089,195
Testifying Expert.
October 2015 – March 2017

• *Free Stream Media Corp. d/b/a Samba TV v. Alphonso, Inc., 3:17-cv-02107 (*California Northern District Court*)*
Testifying Expert
February 2017 – October 2018

• *TicketNetwork, Inc. et al. v. CEATS, Inc.,* Case Nos. IPR2018-0024 and IPR2018-00245 (PTO);
Testifying Expert regarding patent litigation
September 2018 – December 2018

• *CG Technology Development, LLC et al v. DraftKings, Inc.,* Case No. 2:16-cv-00781;
*CG Technology Development, LLC et al v. FanDuel, Inc.*; Case No. 2:16-cv-00801;
*CG Technology Development, LLC et al. v. 888 Holdings PLC*, Case No. 2:16-cv-00856;
*CG Technology Development, LLC et al v. Big Fish Games, Inc.*, Case No. 2:16-cv-00857;
*CG Technology Development, LLC et al v. Double Down Interactive, LLC,* Case No: 2:16-cv-00858;
*CG Technology Development, LLC et al v. Zynga, Inc.,* Case No: 2:16-cv-00859;
*CG Technology Development, LLC et al v. Bwin.Party Digital Entertainment, PPLC et al,* Case No: 2:16-cv-00871(Nevada District Court);
Testifying expert on network gaming, gaming interfaces, and wagering strategies. Performed code reviews and infringement contentions for all defendants.
June 2017 - Current

• *Great West Casualty Company, Bitco General Insurance Corporation and Bitco National Insurance Company v. Intellectual Ventures II, LLC.,* Case NO. IPR2015-01706, U.S. Patent 7,516,177
April 2019 - Current

• *Uniloc v Autodesk*, 2-15-cv-01187 (ED Texas)
Consulting Expert regarding non infringement of US Patent No 7783,523 & 8,515,820 concerning systems that facilitated the pricing of construction projects.
February 2017 – July 2017

• *Bellsouth v American Infoage*, Sago Data Center (Atlanta)
Consulting Expert regarding contract dispute where Bellsouth claims they are entitled to easement / free space and power in Sago Data Center. Providing consulting expertise on telecommunication acts, and LEC / CLEC regulation and Data Center practices.
August 2015

• *Kashless v BuyWithMe*, 2:11-cv-00293-MJP (Seattle)
Consulting Expert regarding eCommerce Coupon Pricing - 6 Patents involved
August 2011

- *Karamelion, LLC. v. AT&T Digital Life, Inc.,* Case No. 2:18-cv-331-JRG (E.D. Tex.)
  Current

- *Rpost Holdings, Inc et al v. DocuSign, Inc.,* Case No: 2:12-cv-00683-JRG (E.D. Tex.)
  January 2019

- *Paid Easy Corp. d/b/a Paideasy v. Mastercard International Incorporated d/b/a Mastercard,* Index No: 63152/2017 (Supreme Court of New York County of Westchester)
  March 2019

- *International Business Machines Corporation (IBM) v. Expedia, Inc.,* Case No: 1:2017-cv-01875-LPS-CJB (Delaware District Court)
  *May 2019 - Current*

- *irth Solutions, LLC v. Apex Data Solutions and Service, LLC et al,* Case No: 6:2018-cv-06884 (New York Western District Court)
  *May 2019 - Current*

- *CG Technology v. William Hill U.S. Holdco, Inc. and Brandy Wine Bookmaking LLC,* Case No: 1:18-cv-00533 (Delaware District Court)
  April 2019 – hold

- *Microchip Technology Inc., v. Delphi Automotive Systems, LLC,* Case No: 17-01194-LPS-CJB (Delaware District Court)
  February 2018 - current

- *Uniloc USA, Inc. et al. v. Motorola Mobility, LLC,* Case No: 17-1658 (Delaware District Court)
  October 2018 – January 2019

- *Ubiquitous Connectivity LP v. TXU Energy Retail Company, LLC,* Case No: 3:2018-02084 (Texas Northern District Court)
  October 2018 - current

- *Digital Media Technologies, Inc. v. Amazon.com, Inc.,* Case No: 0:2017cvpri02409*; Digital Media Technologies, Inc. v. Hulu, LLC,* Case No: 0:2017cvpri02410; *Digital Media Technologies, Inc. v. Netflix, Inc.,* Case No: 0:2017cvpri02408 (U.S. Court of Appeals, Federal Circuit)
  March 2018 - current

- *Implicit, LLC v. Sandvine Corporation,* Case No: 2:2018-cv-00054*; Implicit, LLC v. NETScout Systems, Inc.,* Case No: 2:2018-cv-00053(Texas Eastern District Court)
  October2018

- *Alexsam, Inc. v. Mastercard International Inc.* Case No: 1:2015-cv-02799(New York Eastern District Court)
  *January 2017*- Current
- *SiteLock LLC v. Comodo Holdings Limited et al.,* Case No: 2:2017cv-04080 (Arizona District Court)
  July 2018 – Sept 2018

- *Tamabo, Inc. v. Koninklijke Philips N.V.,* Case No: 2:2017-cv-00750 (Texas Eastern District Court)
  June 2018 – September 2018

- *Energy Intelligence Group, Inc. et al v. Mizuho Bank, Ltd,* Case No:1:17-cv-01508 (Texas Southern District Court)
  July 2018 -August 2018

- *Keynetik, Inc v. Samsung Electronics Co., Ltd and Samsung Electronics America, Inc.,* Case No: 2:2017cv02794 (New Jersey District Court)
  September 2017 – June 2018

- *Edible International, LLC et al v. Google, LLC,* Case No: 3:2018-cv-00216 (Connecticut District Court)
  May 2018

- *Re/Max, LLC v. Quicken Loans, Inc.*, Case No: 1:16-cv-02357(Colorado District Court)
  May 2018

- *Fantasy Interactive Inc. v. HTC Corporation,* Case No: 2:2016-cv-01720 (Washington Western District Court)
  January 2018 – April 2018

- *AGR Group, LLC v. Noble Systems Corp.,* Case No: 8:16-cv-03406 (Florida Middle District Court)
  December 2016 – May 2017

- *Paysys International, Inc v. Atos Se, et al cases,* Case No: 1:2014-cv-10105) New York Southern District Court)
  July 2016 – April 2017

- *IBM v. Amazon.com,* Case No: 9:06CV242 Eastern District of Texas
  *May 2019 - Current*

# Exhibit B – List of Materials

# List of Materials

| Material Title | Document Reference/Source | Bates Number |
|---|---|---|
| Deposition of Christopher Rotsaert - ChristopherJRotsaert_COND 4865-2143-9745 v.1.pdf | Deposition Transcript, Oct 13, 2021 | |
| Deposition of Ben Lo - 7675330 Lo.Ben 120821.miniprint.pdf | Deposition Transcript, Dec 8, 2021 | |
| Deposition of Ben Lo Corporate | Deposition Transcript, Dec 10, 2021 | |
| BBPOS ROAM Engineering and License Agreement | | AC_02770544-554, IngenicoInc_0268234-238 |
| HomeATM-BBPOS License Agreement | | AC_0000917-922 |
| paypal-here-sdk-android-distribution.git | Web download link - https://mvnrepository.com/artifact/com.paypal.retail/here-sdk-debug/2.1.02.19063010 | |
| American National Standard for Financial Services - ANS X9.24-1:2009 - Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques | ANSI Webstore: https://webstore.ansi.org | |
| US8336771 - Tsai - Power Dongle | https://patents.google.com/patent/US8336771B2/en?oq=US8336771 | |
| US8840017 - Chan - Power Management | https://patents.google.com/patent/US8840017B2/en?oq=US8840017+ | |
| US9362689 - Lo - Detect Polarity | https://patents.google.com/patent/US9362689B2/en?oq=US9362689++ | |
| Re iWL - android with Roam Data solution.msg | DISCO Repository | BBPOS_0004382 |
| ROAM Swiper Output Format 10.docx | DISCO Repository | BBPOS_0004383-BBPOS_0004384 |
| Re iWL - android with Roam Data solution.msg | DISCO Repository | BBPOS_0004422-BBPOS_0004423 |
| Re iWL - android with Roam Data solution.msg | DISCO Repository | BBPOS_0004382 |

| Material Title | Document Reference/Source | Bates Number |
|---|---|---|
| SwiperAPI-Android-Guide.doc | DISCO Repository | BBPOS_0004628-BBPOS_0004648 |
| Re ok.msg | DISCO Repository | BBPOS_0005112 |
| BBPOS EMVFlow.docx | DISCO Repository | BBPOS_0005113-BBPOS_0005114 |
| BBPOS TwoWayCommunication.docx | DISCO Repository | BBPOS_0005115-BBPOS_0005116 |
| Re Swiper Track 2 + Track 3.msg | DISCO Repository | BBPOS_0005121-BBPOS_0005122 |
| BBPOS-DataOutputFormat-V1.21.doc | DISCO Repository | BBPOS_0005123-BBPOS_0005137 |
| The documents you requested.msg | DISCO Repository | BBPOS_0005630 |
| Phone list.xlsx | DISCO Repository | BBPOS_0005631 |
| audio interface.pdf | DISCO Repository | BBPOS_0005632 |
| BBPOS-DataOutputFormat-V1.15.doc | DISCO Repository | BBPOS_0005633-BBPOS_0005645 |
| Re One missing scheme for explanation solution to handle the 2 categories of phones for amplitude definition.msg | DISCO Repository | BBPOS_0005646 |
| battery life estimation.pages | DISCO Repository | BBPOS_0005647-BBPOS_0005648 |
| BBPOS-DataOutputFormat-V1.21.doc | DISCO Repository | BBPOS_0005649-BBPOS_0005663 |
| Fwd Paypal G4X - schematic.msg | DISCO Repository | BBPOS_0005664 |
| Paypal-PCB1-ST04-V3.1.pdf | DISCO Repository | BBPOS_0005665-BBPOS_0005667 |
| SwiperDecoder.java (TeresaWongs-Mac-mini-Server's conflicted copy 2012-08-09).svn-base | DISCO Repository | BBPOS_0691264-BBPOS_0691272 |
| Re Format 17 & Format 20 for Track2.msg | DISCO Repository | BBPOS_1632236-BBPOS_1632239 |
| SwiperSimulator.exe | DISCO Repository | BBPOS_1632240 |
| Re Data rate by audio jack2.msg | DISCO Repository | IngenicoInc_0009756 - IngenicoInc_0009757 |
| Re Our confcall next Monday3.msg | DISCO Repository | IngenicoInc_0009883-IngenicoInc_0009891 |
| EMV_SWIPER.pdf | DISCO Repository | IngenicoInc_0010195 - IngenicoInc_0010200 |

Expert Report of Ivan Zatkovich – Trade Secret Misappropriation
Exhibit B – List of Materials

| Material Title | Document Reference/Source | Bates Number |
|---|---|---|
| Re Meeting RoamAPI Ingenico-BBPOS.msg | DISCO Repository | IngenicoInc_0010655 - IngenicoInc_0010656 |
| PRD RP350X _ v5.0 - 03 12 2012.DOC | DISCO Repository | IngenicoInc_0049942 - IngenicoInc_0049966 |
| PRD RP150X _ v2.0 - 07 01 2013.DOC | DISCO Repository | IngenicoInc_0076359 - IngenicoInc_0076380 |
| Re Our confcall next Monday3.msg | DISCO Repository | IngenicoInc_0134751 - IngenicoInc_0134759 |
| EMV_SWIPER.pdf | DISCO Repository | IngenicoInc_0135063 - IngenicoInc_0135068 |
| PRD RP750X _ v4.0 - 31 01 2013.DOC | DISCO Repository | IngenicoInc_0158490 - IngenicoInc_0158525 |
| PRD RP100x DRAFT.DOC | DISCO Repository | IngenicoInc_0190250 - IngenicoInc_0190265 |
| RE Visite BBPOS.msg | DISCO Repository | IngenicoInc_0283863 - IngenicoInc_0283864 |
| rp350x v050.pdf | DISCO Repository | IngenicoInc_0283923-IngenicoInc_0283931 |
| PRD Key Management System _ v1.0.2.DOC | DISCO Repository | IngenicoInc_0079958-IngenicoInc_0079977 |
| RM1&TR1 ITMP with Landi _ 20121029.pptx | DISCO Repository | IngenicoInc_0072949 |
| PRD RP750X _ v7.0 - 04 03 2013.doc | DISCO Repository | IngenicoInc_0181636-IngenicoInc_0181675 |
| Paypal-PCB1-ST04-V1.0.pdf | | BBPOS_1687763-BBPOS_1687765 |
| Swiper-PCB1-ST11-v2.0.pdf | | BBPOS_1687766-BBPOS_1687768 |
| EMVSwiper_PCB1_v0.3.0.pdf | | BBPOS_1687755-BBPOS_1687762 |
| Adaptive threshold flowchart.pdf | | BBPOS_1687849 |
| BBPOS-DataOutputFormat-V1.40.pdf | | BBPOS_1687724-BBPOS_1687754 |

# Exhibit C – Additional References

American National Standard
for Financial Services

# ANS X9.24-1:2009

# Retail Financial Services
# Symmetric Key Management
# Part 1: Using Symmetric Techniques

Secretariat

**Accredited Standards Committee X9, Inc.**

Approved: October 13, 2009

**American National Standards Institute**

# Foreword

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretation should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

**CAUTION NOTICE:** This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval.

Published by

**Accredited Standards Committee X9, Incorporated**
**Financial Industry Standards**
**1212 West Street, Suite 200**
**Annapolis, MD 21401 USA**
**X9 Online http://www.x9.org**

# Contents

## Figures

# Tables

# Introduction

Today, billions of dollars in funds are transferred electronically by various communication methods. Transactions are often entered remotely, off-premise from financial institutions, by retailers or by customers directly. Such transactions are transmitted over potentially non-secure media. The vast range in value, size, and the volume of such transactions expose institutions to severe risks, which may be uninsurable.

To protect these financial messages and other sensitive information, many institutions are making increased use of the American National Standards Institute Triple Data Encryption Algorithm (TDEA). Specific examples of its use include standards for message authentication, personal identification number encryption, other data encryption, and key encryption.

The TDEA is in the public domain. The security and reliability of any process based on the TDEA is directly dependent on the protection afforded to secret numbers called cryptographic keys. This part of this standard deals exclusively with management of symmetric keys using symmetric techniques. ANS X9.24-2 addresses management of symmetric keys using asymmetric techniques.

A familiar analogy may be found in the combination lock of a vault. The lock design is public knowledge. Security is provided by keeping a number, the combination, a secret. Secure operation also depends on protective procedures and features which prevent surreptitious viewing or determination of the combination by listening to its operation. Procedures are also required to ensure that the combination is random and cannot be modified by an unauthorized individual without detection.

Suggestions for the improvement of this standard will be welcome. They should be sent to the ASC X9 Secretariat, Accredited Standards Committee X9, Inc., 1212 West Street, Suite 200, Annapolis, MD 21401.

The standard was processed and approved for submittal to the American National Standards Institute by the Accredited Standards Committee X9 - Financial Services. Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X9 Committee had the following members:

Roy DeCicco, X9 Chairman
Vincent DeSantis, X9 Vice-Chairman
Cynthia Fuller, Executive Director
Janet Busch, Program Manager

| Organization Represented | Representative | |
|---|---|---|
| ACI Worldwide | Doug | Grote |
| ACI Worldwide | Cindy | Rink |
| American Bankers Association | Tom | Judd |
| American Bankers Association | C. Diane | Poole |
| American Express Company | Ted | Peirce |
| Apriva | Len | Sutton |
| Bank of America | Andi | Coleman |
| Bank of America | Daniel | Welch |
| Certicom Corporation | Daniel | Brown |
| Citigroup, Inc. | Mark | Clancy |
| Citigroup, Inc. | Michael | Knorr |

| | | |
|---|---|---|
| Citigroup, Inc. | Karla | McKenna |
| Citigroup, Inc. | Chii-Ren | Tsai |
| CUSIP Service Bureau | Gerard | Faulkner |
| CUSIP Service Bureau | James | Taylor |
| Deluxe Corporation | John | FitzPatrick |
| Deluxe Corporation | Ralph | Stolp |
| Diebold, Inc. | Bruce | Chapa |
| Diebold, Inc. | Anne | Konecny |
| Discover Financial Services | Dave | Irwin |
| Discover Financial Services | Deana | Morrow |
| Federal Reserve Bank | Deb | Hjortland |
| Federal Reserve Bank | Claudia | Swendseid |
| First Data Corporation | Todd | Nuzum |
| First Data Corporation | Rick | Van Luvender |
| Fiserv | Bud | Beattie |
| Fiserv | Kevin | Finn |
| Fiserv | Lori | Hood |
| Fiserv | Dan | Otten |
| Fiserv | Skip | Smith |
| FIX Protocol Ltd | Jim | Northey |
| Harland Clarke | John | McCleary |
| Hewlett Packard | Larry | Hines |
| Hewlett Packard | Gary | Lefkowitz |
| IBM Corporation | Todd | Arnold |
| IFSA | Dexter | Holt |
| IFSA | Dan | Taylor |
| Ingenico | Steve | McKibben |
| Ingenico | John | Spence |
| J.P. Morgan Chase & Co | Robert | Blair |
| J.P. Morgan Chase & Co | Roy | DeCicco |
| J.P. Morgan Chase & Co | Edward | Koslow |
| J.P. Morgan Chase & Co | Jackie | Pagan |
| J.P. Morgan Chase & Co | Charita | Wamack |
| Key Innovations | Scott | Spiker |
| Key Innovations | Paul | Walters |
| KPMG LLP | Mark | Lundin |
| MagTek, Inc. | Terry | Benson |
| MagTek, Inc. | Jeff | Duncan |
| MagTek, Inc. | Mimi | Hart |
| MasterCard International | Mark | Kamers |
| Merchant Advisory Group | Dodd | Roberts |
| Metavante Image Solutions | Stephen | Gibson-Saxty |
| NACHA The Electronic Payments Association | Nancy | Grant |
| National Association of Convenience Stores | Michael | Davis |
| National Association of Convenience Stores | Alan | Thiemann |
| National Security Agency | Paul | Timmel |
| NCR Corporation | David | Norris |
| NCR Corporation | Steve | Stevens |
| RouteOne | Mark | Leonard |
| SWIFT/Pan Americas | Jean- | Eloy |

| | | |
|---|---|---|
| | Marie | |
| SWIFT/Pan Americas | James | Wills |
| SWIFT/Pan Americas | Jamie | Shay |
| TECSEC Incorporated | Ed | Scheidt |
| The Clearing House | Vincent | DeSantis |
| U.S. Bank | Brian | Fickling |
| U.S. Bank | Gregg | Walker |
| University Bank | Stephen | Ranzini |
| University Bank | Michael | Talley |
| VeriFone, Inc. | David | Ezell |
| VeriFone, Inc. | Dave | Faoro |
| VeriFone, Inc. | Allison | Holland |
| VeriFone, Inc. | Doug | Manchester |
| VeriFone, Inc. | Brad | McGuinness |
| VeriFone, Inc. | Brenda | Watlington |
| VISA | Brian | Hamilton |
| VISA | John | Sheets |
| VISA | Richard | Sweeney |
| Wells Fargo Bank | Andrew | Garner |
| Wells Fargo Bank | Mike | McCormick |
| Wells Fargo Bank | Mike | Rudolph |
| Wells Fargo Bank | Mark | Tiggas |
| Wincor Nixdorf Inc | Ramesh | Arunashalam |
| XBRL US, Inc. | Mark | Bolgiano |

At the time it approved this standard, the X9F Subcommittee on Data and Information Security had the following members:

Dick Sweeney, Chairperson

| **Organization Represented** | **Representative** | |
|---|---|---|
| Company | First Name | Last Name |
| ACI Worldwide | Doug | Grote |
| ACI Worldwide | Julie | Samson |
| ACI Worldwide | Sid | Sidner |
| American Bankers Association | Tom | Judd |
| American Express Company | William J. | Gray |
| American Express Company | Vicky | Sammons |
| Bank of America | Dion | Bellamy |
| Bank of America | Terrelle | Carswell |
| Bank of America | Andi | Coleman |

| | | |
|---|---|---|
| Bank of America | Todd | Inskeep |
| Bank of America | John | McGraw |
| Bank of America | Chris | Schrick |
| Bank of America | Daniel | Welch |
| Certicom Corporation | Daniel | Brown |
| Certicom Corporation | John O. | Goyo |
| Certicom Corporation | Sandra | Lambert |
| Certicom Corporation | Scott | Vanstone |
| Citigroup, Inc. | Mark | Clancy |
| Citigroup, Inc. | Susan | Rhodes |
| Citigroup, Inc. | Gary | Word |
| Communications Security Establishment | Alan | Poplove |
| Communications Security Establishment | Bridget | Walshe |
| Cryptographic Assurance Services | Ralph | Poore |
| Cryptographic Assurance Services | Jeff | Stapleton |
| CUSIP Service Bureau | Scott | Preiss |
| CUSIP Service Bureau | James | Taylor |
| DeLap LLP | Steve | Case |
| DeLap LLP | Darlene | Kargel |
| Deluxe Corporation | John | FitzPatrick |
| Deluxe Corporation | Ralph | Stolp |
| Depository Trust and Clearing Corporation | Robert | Palatnick |
| Diebold, Inc. | Bruce | Chapa |
| Diebold, Inc. | Laura | Drozda |
| Diebold, Inc. | Scott | Harroff |
| Diebold, Inc. | Anne | Konecny |
| Diebold, Inc. | Jessica | Wapole |

| | | |
|---|---|---|
| Discover Financial Services | Julie | Shaw |
| Entrust, Inc. | Sharon | Boeyen |
| Entrust, Inc. | Miles | Smid |
| Federal Reserve Bank | Darin | Contini |
| Federal Reserve Bank | Pieralberto | Deganello |
| Federal Reserve Bank | Deb | Hjortland |
| Federal Reserve Bank | Mike | Ram |
| Ferris and Associates, Inc. | J. Martin | Ferris |
| First Data Corporation | Lisa | Curry |
| First Data Corporation | Lilik | Kazaryan |
| First Data Corporation | Todd | Nuzum |
| First Data Corporation | Scott | Quinn |
| First Data Corporation | Andrea | Stallings |
| First Data Corporation | Rick | Van Luvender |
| Fiserv | Bud | Beattie |
| Fiserv | Mary | Bland |
| Fiserv | Kevin | Finn |
| Fiserv | Dennis | Freiburg |
| Fiserv | Dan | Otten |
| Futurex | Greg | Schmid |
| GEOBRIDGE Corporation | Jason | Way |
| Harland Clarke | Joseph | Filer |
| Harland Clarke | John | McCleary |
| Harland Clarke | John | Petrie |
| Heartland Payment Systems | Roger | Cody |
| Heartland Payment Systems | Glenda | Preen |
| Hewlett Packard | Larry | Hines |

| | | |
|---|---|---|
| Hewlett Packard | Susan | Langford |
| Hewlett Packard | Gary | Lefkowitz |
| Hypercom | Mohammad | Arif |
| Hypercom | Gary | Zempich |
| IBM Corporation | Todd | Arnold |
| IBM Corporation | Michael | Kelly |
| IFSA | Dexter | Holt |
| Independent Community Bankers of America | Cary | Whaley |
| InfoGard Laboratories | Doug | Biggs |
| InfoGard Laboratories | Ken | Kolstad |
| Ingenico | John | Spence |
| J.P. Morgan Chase & Co | Robert | Blair |
| J.P. Morgan Chase & Co | Edward | Koslow |
| J.P. Morgan Chase & Co | Kathleen | Krupa |
| J.P. Morgan Chase & Co | Donna | Meagher |
| J.P. Morgan Chase & Co | Jackie | Pagan |
| K3DES LLC | Azie | Amini |
| Key Innovations | Scott | Spiker |
| KPMG LLP | Mark | Lundin |
| MagTek, Inc. | Terry | Benson |
| MagTek, Inc. | Jeff | Duncan |
| MagTek, Inc. | Mimi | Hart |
| MasterCard International | Jeanne | Moore |
| MasterCard International | Michael | Ward |
| Merchant Advisory Group | Brad | Andrews |
| Merchant Advisory Group | Dodd | Roberts |
| National Institute of Standards and Technology | Elaine | Barker |

| | | |
|---|---|---|
| National Institute of Standards and Technology | Lawrence | Bassham III |
| National Institute of Standards and Technology | William | Burr |
| National Institute of Standards and Technology | Lily | Chen |
| National Institute of Standards and Technology | David | Cooper |
| National Institute of Standards and Technology | Morris | Dworkin |
| National Institute of Standards and Technology | Randall | Easter |
| National Institute of Standards and Technology | Sharon | Keller |
| National Institute of Standards and Technology | John | Kelsey |
| National Institute of Standards and Technology | Annabelle | Lee |
| National Institute of Standards and Technology | Fernando | Podio |
| National Security Agency | Mike | Boyle |
| National Security Agency | Greg | Gilbert |
| National Security Agency | Tim | Havighurst |
| National Security Agency | Paul | Timmel |
| National Security Agency | Debby | Wallner |
| NCR Corporation | Charlie | Harrow |
| NCR Corporation | Ali | Lowden |
| NCR Corporation | David | Norris |
| NCR Corporation | Ron | Rogers |
| NCR Corporation | Steve | Stevens |
| NCR Corporation | Ally | Whytock |
| NTRU Cryptosystems, Inc. | Nick | Howgrave-Graham |

| | | |
|---|---|---|
| NTRU Cryptosystems, Inc. | Ari | Singer |
| NTRU Cryptosystems, Inc. | William | Whyte |
| Pitney Bowes, Inc. | Andrei | Obrea |
| Pitney Bowes, Inc. | Leon | Pintsov |
| Pitney Bowes, Inc. | Rick | Ryan |
| RBS Group | Dan | Collins |
| Rosetta Technologies | Jim | Maher |
| Rosetta Technologies | Paul | Malinowski |
| RSA, The Security Division of EMC | Steve | Schmalz |
| Surety, Inc. | Dimitrios | Andivahis |
| Surety, Inc. | Tom | Klaff |
| TECSEC Incorporated | Ed | Scheidt |
| TECSEC Incorporated | Dr. Wai | Tsang |
| TECSEC Incorporated | Jay | Wack |
| Thales e-Security, Inc. | Colette | Broadway |
| Thales e-Security, Inc. | Jose | Diaz |
| Thales e-Security, Inc. | Tim | Fox |
| Thales e-Security, Inc. | James | Torjussen |
| The Clearing House | Vincent | DeSantis |
| The Clearing House | Henry | Farrar |
| The Clearing House | Susan | Long |
| U.S. Bank | Glenn | Marshall |
| U.S. Bank | Peter | Skirvin |
| U.S. Bank | Robert | Thomas |
| Unisys Corporation | David J. | Concannon |
| Unisys Corporation | Navnit | Shah |
| University Bank | Stephen | Ranzini |

| | | |
|---|---|---|
| University Bank | Michael | Talley |
| VeriFone, Inc. | John | Barrowman |
| VeriFone, Inc. | David | Ezell |
| VeriFone, Inc. | Dave | Faoro |
| VeriFone, Inc. | Doug | Manchester |
| VeriFone, Inc. | Brad | McGuinness |
| VeriFone, Inc. | Brenda | Watlington |
| VISA | Leon | Fell |
| VISA | Tara | Kissoon |
| VISA | Chackan | Lai |
| VISA | Stoddard | Lambertson |
| VISA | Chris | McDaniel |
| VISA | John | Sheets |
| VISA | Richard | Sweeney |
| VISA | Johan (Hans) | Van Tilburg |
| Voltage Security, Inc. | Luther | Martin |
| Voltage Security, Inc. | Terence | Spies |
| Wells Fargo Bank | Mick | Bauer |
| Wells Fargo Bank | Jason | Buck |
| Wells Fargo Bank | Andrew | Garner |
| Wells Fargo Bank | Jeff | Jacoby |
| Wells Fargo Bank | Brian | Keltner |
| Wells Fargo Bank | Israel | Laracuente |
| Wells Fargo Bank | Eric | Lengvenis |
| Wells Fargo Bank | Mike | McCormick |
| Wells Fargo Bank | David | Naelon |
| Wells Fargo Bank | Doug | Pelton |

| | | |
|---|---|---|
| Wells Fargo Bank | Chuck | Perry |
| Wells Fargo Bank | Keith | Ross |
| Wells Fargo Bank | Mike | Rudolph |
| Wells Fargo Bank | Ruven | Schwartz |
| Wells Fargo Bank | Craig | Shorter |
| Wells Fargo Bank | Tony | Stieber |
| Wincor Nixdorf Inc | Ramesh | Arunashalam |
| Wincor Nixdorf Inc | Saul | Caprio |
| Wincor Nixdorf Inc | Joerg-Peter | Dohrs |
| Wincor Nixdorf Inc | Matthias | Runowski |
| Wincor Nixdorf Inc | Adam | Sandoval |
| Wincor Nixdorf Inc | Michael | Waechter |

The X9F6 working group that revised this standard consisted of the following members:

John Sheets, Chairperson

| Organization Represented | Representative | |
|---|---|---|
| ACI Worldwide | Doug | Grote |
| ACI Worldwide | Jim | Jeter |
| ACI Worldwide | Sid | Sidner |
| Bank of America | Andi | Coleman |
| DeLap LLP | Steve | Case |
| DeLap LLP | Darlene | Kargel |
| Diebold, Inc. | Bruce | Chapa |
| Dresser Wayne | Tim | Weston |
| Fagan and Associates, LLC | Jeanne | Fagan |
| First Data Corporation | Lisa | Curry |
| First Data Corporation | Lilik | Kazaryan |
| First Data Corporation | Scott | Quinn |
| First Data Corporation | Andrea | Stallings |
| Fiserv | Dan | Otten |
| Futurex | Chris | Hamlett |
| GEOBRIDGE Corporation | Jason | Way |
| Gilbarco | Bruce | Welch |
| Heartland Payment Systems | Roger | Cody |
| Heartland Payment Systems | Glenda | Preen |
| Hewlett Packard | Larry | Hines |
| Hypercom | Gary | Zempich |
| IBM Corporation | Todd | Arnold |
| Ingenico | John | Spence |

| | | |
|---|---|---|
| K3DES LLC | James | Richardson |
| K3DES LLC | Azie | Amini |
| Key Innovations | Scott | Spiker |
| Mustang Microsystems, Inc. | Tom | Galloway |
| NCR Corporation | Charlie | Harrow |
| RP Kastner Consulting, Inc. | Rick (Richard P.) | Kastner |
| SafeNet, Inc. | Brett | Thompson |
| Thales e-Security, Inc. | Jose | Diaz |
| Thales e-Security, Inc. | James | Torjussen |
| VeriFone, Inc. | Doug | Manchester |
| VISA | John | Sheets |
| Wells Fargo Bank | Craig | Shorter |

# Retail Financial Services
# Symmetric Key Management
# Part 1: Using Symmetric Techniques

## 1 Purpose

This key management standard, utilized in conjunction with the American National Standard Triple Data Encryption Algorithm (TDEA) (see Reference 2), is used to manage symmetric keys that can be used to protect messages and other sensitive information in a financial services environment. The security and reliability of any process based on the TDEA is directly dependent on the protection afforded to secret parameters called cryptographic keys.

This standard establishes requirements and guidelines for the secure management and application-level interoperability of keying operations. Such keys could be used for authenticating messages (see Reference 5), for encrypting Personal Identification Numbers (PIN) (see Reference 4), for encrypting other data, and for encrypting other keys.

## 2 Scope

This part of this standard covers both the manual and automated management of keying material used for financial services such as point-of-sale (POS) transactions (debit and credit), automated teller machine (ATM) transactions, messages among terminals and financial institutions, and interchange messages among acquirers, switches and card issuers. This part of this standard deals exclusively with management of symmetric keys using symmetric techniques. This part of this standard specifies the minimum requirements for the management of keying material. Addressed are all components of the key management life cycle including generation, distribution, utilization, storage, archiving, replacement and destruction of the keying material. An institution's key management process, whether implemented in a computer or a terminal, is not to be implemented or controlled in a manner that has less security, protection, or control than described herein. It is intended that two nodes, if they implement compatible versions of:

- the same secure key management method,
- the same secure key identification technique approved for a particular method, and
- the same key separation methodologies

in accordance with this part of this standard will be interoperable at the application level. Other characteristics may be necessary for node interoperability; however, this part of this standard does not cover such characteristics as message format, communications protocol, transmission speed, or device interface.

### 2.1 Application

This part of this standard is applicable for institutions implementing techniques to safeguard cryptographic keys used for authentication and encryption of messages and other sensitive data. Specifically, this applies to institutions in the financial services industry implementing References 4 and/or 5.

Mandatory standard techniques and procedures are indicated by the word '**SHALL**'. Guidelines are indicated by the word '**SHOULD**'.

## 3 References

This part of this standard shall be used in conjunction with the following publications.

1. ANS INCITS 92-1981 (R2003), Information Technology - Data Encryption Algorithm (DEA)
2. NIST SP800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
3. ANS X9.65-2004, Triple Data Encryption Algorithm (TDEA) Implementation

The following publications are applicable and may be referenced in this part of this standard.

4. ANS X9.8-1-2003, Personal Identification Number (PIN) Management and Security
5. ISO 16609, Banking – Requirements for Message Authentication using Symmetric Techniques
6. ISO 7812-1985 , Identification cards – Numbering system and registration procedure for issuer identifiers
7. ISO 8583-1993, Bankcard Originated Messages – Interchange Message Specifications – Content for Financial Transactions
8. ISO 9797-1, Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher

**The versions listed were current as of the publication of this document, however these documents are routinely updated and reaffirmed. The current versions SHOULD be referenced when using this part of this standard.**

## 4 Terms and Definitions

**4.1**
**acceptor**
same as “card acceptor”

**4.2**
**acquirer**
the institution (or its agent) which acquires from the card acceptor the financial data relating to the transaction and initiates that data into an interchange system

**4.3**
**algorithm**
a clearly specified mathematical process for computation; a set of rules which, if followed, will give a prescribed result

**4.4**
**archived key**
an inactive key that is being saved in a secure manner for a non-operational purpose such as a legal requirement for future recovery

**4.5**
**authentication**
the act of determining that a message has not been changed since leaving its point of origin. The identity of the originator is implicitly verified

**4.6**
**authentication algorithm**
the application of a cryptographic process in which output text depends on all preceding input text

**4.7**
**authentication element**
a contiguous group of bits or characters which are to be protected by being processed by the authentication algorithm

**4.8**
**base derivation key**
a derivation key normally associated with Derived Unique Key Per Transaction

**4.9**
**card acceptor**
party accepting the card and presenting transaction data to the acquirer

**4.10**
**card issuer**
the institution or its agent that issues the card to the cardholders

**4.11**
**check value**
a computed value which is the result of passing a data value through a non-reversible algorithm

**4.12**
**ciphertext**
data in its enciphered form

**4.13**
**cleartext**
data in its original, unencrypted form

**4.14**
**communicating pair**
two entities (usually institutions) sending and receiving transactions. This is to include alternate processing sites either owned or contracted by either communicating entity

**4.15**
**compromise**
in cryptography, the breaching of secrecy and/or security. A violation of the security of a system such that an unauthorized disclosure of sensitive information may have occurred

**4.16**
**cryptographic key**
a parameter that determines the operation of a cryptographic function such as:

a) the transformation from cleartext to ciphertext and vice versa

b) synchronized generation of keying material

c) digital signature computation or validation

**4.17**
**cryptographic key synchronization**
the ability for two nodes, that cryptographically process a transaction, to determine the identical Transaction Key

**4.18**
**Data Encryption Algorithm (DEA)**
the cryptographic algorithm adopted by ANSI (see Reference 1)

**4.19**
**decryption**
a process of transforming ciphertext (unreadable) into cleartext (readable)

**4.20**
**derivation key**
a key which is used to compute cryptographically another key. Normally a single derivation key is used in a transaction-receiving (e.g., acquirer) TRSM to derive or decrypt the Transaction Keys used by a large number of originating (e.g., terminal) TRSMs

**4.21**
**double-length key**
a TDEA key having a length of 128 bits (see reference 2 'keying option 2' and reference 3 'TDEA 2-key implementation')

**4.22**
**dual control**
a process of utilizing two or more separate entities (usually persons), operating in concert, to protect sensitive functions or information. Both entities are equally responsible for the physical protection of materials involved in vulnerable transactions. It **SHALL** be ensured that no one entity is able to access or to utilize the materials (e.g., cryptographic key). For manual key generation, conveyance, loading, storage and retrieval, dual control requires split knowledge of keys among the entities. Also see "split knowledge"

**4.23**
**DUKPT**
Derived Unique Key per Transaction - a key management method which uses a unique key for each transaction, and prevents the disclosure of any past key used by the transaction-originating TRSM. The unique Transaction Keys are derived from a base derivation key using only non-secret data transmitted as part of each transaction

**4.24**
**encryption**
a process of transforming cleartext (readable) into ciphertext (unreadable) for the purpose of security or privacy

**4.25**
**exclusive-or**
a mathematical operation, symbol "XOR", defined as:
0 XOR 0 = 0  0 XOR 1 = 1  1 XOR 0 = 1  1 XOR 1 = 0
Equivalent to binary addition without carry (modulo-2 addition)

**4.26**
**institution**
an establishment responsible for facilitating customer initiated transactions or transmission of funds for the extension of credit, or the custody, loan, exchange, or issuance of money

**4.27**
**interchange**
mutual acceptance and exchange of messages between financial institutions

**4.28**
**issuer**
the institution holding the account identified by the primary account number (PAN)

**4.29**
**key**
see cryptographic key

**4.30**
**key component**
one of at least two parameters having the format of a cryptographic key that is exclusive-ored/added modulo-2 with one or more like parameters to form a cryptographic key. A component is equal in length to the resulting key

**4.31**
**key encrypting key**
a key used exclusively to encrypt and decrypt keys

**4.32**
**keying material**
the data (e.g., keys and initialization vectors) necessary to establish and maintain cryptographic keying relationships

**4.33**
**key separation**
a process for ensuring that a key is used for only its intended purpose

**4.34**
**key set**
a group of keys all determined by a common cryptographic procedure and differentiated by non-secret input to this procedure such that knowledge of one key does not disclose any other key in the group

**4.35**
**key set identifier**
a non-secret value which uniquely identifies a key set

**4.36**
**master key**
in a hierarchy of Key Encrypting Keys and Transaction Keys, the highest level of Key Encrypting Key is known as a Master Key

**4.37**
**message**
a communication containing one or more transactions or related information

**4.38**
**message authentication code (MAC)**
a cryptographic value which is the result of passing a financial message through the message authentication algorithm using a specific key

**4.39**
**node**
any point in a network that does some form of processing of data, such as a terminal, acquirer or switch

**4.40**
**non-reversible transformation**
encryption of cleartext in such a way that the ciphertext cannot be decrypted back to the original cleartext

**4.41**
**Originator**
the person, institution or other entity that is responsible for and authorized to originate a message

**4.42**
**parity**
a measure of the number of '1' bits in a group of '0' and '1' bits; either odd or even

**4.43**
**privacy**
the confidential nature of data which requires protection against unauthorized disclosure

**4.44**
**pseudo random**
a value which is statistically random and essentially unpredictable although generated by an algorithm

**4.45**
**random**
a value in a set that has equal probability of being selected from the total population of possibilities, hence unpredictable

**4.46**
**recipient**
the person, institution or other entity that is responsible for and authorized to receive a message

**4.47**
**replay**
the process of sending a previously sent message as a method of perpetrating a fraud

**4.48**
**sender**
the person, institution, or other entity transmitting a message

**4.49**
**single length key**
a cryptographic key having a length of 56 bits plus 8 parity bits

**4.50**
**SMID**
Security Management Information Data element used to manage and control cryptographic operations

**4.51**
**split knowledge**
a condition under which two or more parties separately and confidentially have information (e.g., key components) which, individually, convey no knowledge of the resulting combined information (e.g., cryptographic key)

**4.52**
**symmetric key**
a cryptographic key that is used in a symmetric cryptographic algorithm (e.g., TDEA). The same symmetric key that is used for encryption is also used for decryption

**4.53**
**switch**
a node that can route data from a node to other nodes

**4.54**
**tampering**
the penetration or modification of internal operation and/or insertion of active or passive tapping mechanisms to determine or record secret data

**4.55**
**TDEA**
see "Triple Data Encryption Algorithm"

**4.56**
**terminal**
a device/system that initiates a transaction

**4.57**
**transaction**
a series of messages to perform a predefined function

**4.58**
**transaction key**
a key used to cryptographically process the transaction. If more than one key is used for different cryptographic functions, each may be a variant of the Transaction Key. A Transaction Key is sometimes referred to as a Data Key, communications key, session key, or working key

**4.59**
**Triple Data Encryption Algorithm**
the algorithm specified in Reference 2.

**4.60**
**triple-length key**
a TDEA key having a length of 192 bits (see reference 2 'keying option 1' and reference 3 'TDEA 3-key implementation')

**4.61**
**TRSM**
Tamper Resistant Security Module (see Section 7.2)

**4.62**
**UKPT**
Unique Key Per Transaction

**4.63**
**variant of a key**
a new key formed by a process (which need not be secret) with the original key, such that one or more of the non-parity bits of the new key differ from the corresponding bits of the original key

**4.64**
**verification**
the process of associating and/or checking a unique characteristic

**4.65**
**XOR**
see "exclusive-or"

# 5 Standard Organization

The remainder of this part of this standard is divided into three sections.

Section 6 describes the environment in which key management techniques operate

Section 7 establishes the requirements applicable to key management

Section 8 specifies key management methods, key identification techniques, and the additional requirements for each specific method

# 6 Environment

## 6.1 General

In order to provide insight into key management requirements, this section provides a generic description of each of the entities involved. Transaction processing systems are composed of subsystems operated by one or more of these entities. The method of identification in such systems has considerable effect on the overall security requirements. While this part of this standard primarily addresses transaction processing systems (e.g., ATM systems, POS networks) using magnetic cards as part of the identification process, other applications such as home banking, smart cards, and wholesale corporate banking are not precluded.

## 6.2 Cardholder and Card Issuer

The card issuer guarantees payment to the acquirer for transactions or services rendered to the cardholder upon proper authentication and authorization of the cardholder, provided all required conditions are met. The card serves to identify the cardholder and the card issuer. In addition, the card may carry other information such as period of validity (e.g., expiration date) and security related information (e.g., PIN offset).

The cardholder and issuer may agree on a secret Personal Identification Number (PIN) (see Reference 4) to be used during transactions. The transaction processing system has an obligation to maintain the PIN secrecy while transporting the transaction from cardholder to card issuer. Note that the card issuer may delegate responsibility for verification of the PIN to an agent.

## 6.3 Card Acceptor

The card acceptor accepts cards to access the cardholders' account(s) or as a means of payment for goods or services. In POS systems this may be a retailer, service company, financial institution, etc.; in ATM systems the card acceptor may be the same party as the acquirer. Rather than accepting the card as direct proof of payment, the acceptor may forward transaction information to an acquirer. The acceptor will take a transaction authorization from the acquirer as guarantee for payment. The security of the transaction information exchanged with the acquirer is important. Security features may include message authentication (see Reference 5), secrecy of the PIN (see Reference 4), etc.

### 6.4 Acquirer

The acquirer provides transaction processing to acceptors. For some transactions the acquirer may authorize a transaction acting as an agent of an issuer. In other cases (e.g., the transaction value exceeds a certain threshold) the transaction information is sent to an issuer or its agent for authorization.

For the acquisition function the acquirer needs facilities that provide secure processing for translation of PINs in node-to-node systems, message authentication for transaction exchanges, etc. For combined acquisition and authorization functions, the acquirer needs security facilities to satisfy the requirements of the issuer it represents.

## 7 Key Management Requirements

### 7.1 General

a) Cryptographic keys **SHALL** only exist in one or more of the following forms:

   1) In a Tamper-Resistant Security Module (TRSM) as specified in Section 7.2 below.

   2) If outside a TRSM, as a cryptogram that **SHALL** have been created inside a TRSM by TDEA using a Key Encrypting Key.

   3) If non-encrypted and outside of a TRSM, a key **SHALL** exist only in one of the following forms:

      i) as two or more key components as defined in Section 7.5, employing dual control and split knowledge or

      ii) as a cleartext key while being transferred from a Key Loading Device (KLD) to a directly connected TRSM under dual control in an environment meeting the requirements of 7.5.

b) Any key resident in a transaction-originating terminal **SHALL** exist only in that device and those facilities which are authorized to receive and/or transmit encrypted data or authentication elements from or to that device.

c) Compromise of a shared key between entities **SHALL** not compromise keys shared with other entities.

d) An unauthorized attempt to enter a key or restore a replaced key **SHALL** be precluded or detected.

e) Cryptographic keys **SHALL** be protected against both disclosure and misuse.

f) Unauthorized modification, substitution, or replay of a key **SHALL** be prevented or detected.

g) Cryptographic keys may be triple length, but **SHALL** be at least double length keys (see Reference 2 keying options 1 and 2).

h) For all TDEA modes of operation, the three cryptographic keys (K1, K2, K3) define a TDEA key bundle (see Reference 2 keying options 1 and 2). Note when using double length keys, K1 = K3. The bundle and the individual keys **SHALL**:

1) be secret;

2) be generated randomly or pseudo-randomly;

3) have integrity whereby each key in the bundle has not been altered in an unauthorized manner since the time it was generated, transmitted, or stored by an authorized source;

4) be used in the appropriate order as specified by the particular mode;

5) be considered a fixed quantity in which an individual key cannot be manipulated while leaving the other two keys unchanged; and

6) cannot be unbundled for any purpose.

i) All instances for which cryptographic keys are created, received, accessed, distributed, stored, or destroyed, are recorded, including any hardware used in support of these activities.

j) A key encrypting key **SHALL** be the same or greater cryptographic strength as the key that it is encrypting.

k) Cryptographic keys **SHALL** exist in the least number of locations consistent with the operation of the system.

## 7.2 Tamper-Resistant Security Modules (TRSM) used for Key Management

A Tamper-Resistant Security Module (TRSM) used for key management is a device with physical characteristics that make successful tampering difficult and improbable. A TRSM **SHALL** have physical characteristics that inhibit the determination of any secret data including any past, present, or future keys. A TRSM **SHALL** have physical and functional (logical) characteristics that, in combination, preclude the determination of any key previously used by the device to encrypt or decrypt secret data.

To preclude the determination of any key previously used by the device to encrypt or decrypt secret data, the TRSM **SHALL** use one or both of the following methods, in combination with appropriate security procedures:

— Physical barriers

— Unique key per transaction.

All TRSMs **SHALL** have features that resist successful tampering. Tampering includes penetration without zeroization of security parameters, unauthorized modification of the TRSM's internal operation, or insertion of tapping mechanisms or non-intrusive eavesdropping methods to determine, record, or modify secret data. Such features **SHALL** include one or more of the following:

— The TRSM includes means that detect attempted tampering and thereupon cause the automatic erasure of all cleartext keying material contained in the device. Tamper detection **SHALL** be active regardless of the power state of the TRSM.

— The TRSM is constructed with physical barriers that make successful tampering infeasible.

— The TRSM is sufficiently resistant to tampering that successful tampering requires an extended time, such that the absence of the TRSM from its authorized location, or its subsequent return to this location, has a high probability of being noted before the device is (again) used for cryptographic operations.

- The TRSM is constructed in such a way that successful tampering causes visible damage to the device that has a high probability of being noted after the device has been returned to its authorized location but before it is (again) used for cryptographic operations. However, when a TRSM employs this feature exclusively, it **SHALL** be used only in conjunction with unique key per transaction key management.

- The TRSM is constructed in such a way that it is not feasible to modify individual or groups of bits in keys stored in the TRSM. (The only method available to modify a key is to replace it in its entirety.)

In addition, all TRSMs **SHALL** prevent the disclosure of any key that has been used to encrypt or decrypt secret data, including other keys. TRSMs that retain any such key require "compromise prevention". Such a TRSM **SHALL** be designed to be tamper resistant by employing physical barriers so that there is a negligible probability of tampering that could successfully disclose such a key. TRSMs that do not retain any such key require only "compromise detection", and may be less tamper resistant. Compromise of a key resident in such a TRSM does not disclose previously encrypted data, but it is necessary to prevent the future use of any such key in the event that the TRSM is suspected of being compromised. Since any key that might be disclosed by the compromise has not yet been used, it is only necessary to ensure that this key is never used (except by chance). In general it is easier to detect a compromise (and then prevent subsequent use of the compromised TRSM and its keys) than to prevent a compromise.

A TRSM **SHALL** provide compromise prevention, and employ physical barriers, unless both of the following requirements are met:

- The TRSM employs a unique-key-per-transaction technique with the characteristic that no information residing in the TRSM at the completion of a transaction discloses (even with the knowledge of additional relevant data that is, or has been, available external to the TRSM) any key that the device has used to encrypt or decrypt secret data during or preceding the transaction.

- Should the TRSM be compromised and its keys ascertained, these keys cannot be loaded into another identical (e.g., stolen) TRSM. (For example, this objective can be achieved if the key loading process internal to the TRSM involves non-reversible transformations).

A TRSM may be combined with one or more physically separated TRSMs and still be considered as a single TRSM provided that all TRSMs and the transmission medium connecting them meet the above requirements.

The process of placing an initial key into a TRSM **SHALL** establish a high degree of confidence that the key is being transferred into a TRSM which is legitimate, and which has not been modified to permit the future unauthorized disclosure of keying material and/or data to be protected by such keying material. Furthermore, the process for loading an initial key into a TRSM **SHALL** assure the secrecy of this key. It **SHALL** not be possible for any person to ascertain all or part of any final key during the initial key distribution process. However, the key may be loaded as two or more key components as in Section 7.5.

When not in active use, a TRSM (especially a terminal's TRSM) **SHALL** be stored in an environment which minimizes the probability of unauthorized modifications such as the insertion of an active or passive tapping mechanism. The TRSM, when containing a key, **SHALL** be stored in a controlled environment where there is a minimum risk that the TRSM might be stolen, modified in an unauthorized way, and then returned to storage without detection. When a TRSM is removed for repair or permanently removed from service (excluding lost or stolen devices, see section 7.7), all keys **SHALL** be erased from the TRSM as per section 7.8.

Every node participating in key management **SHALL** contain a TRSM.

## 7.3 A Secure Environment

A secure environment is protected physically, procedurally and logically to prevent the disclosure of any keying material existing within the environment. A secure environment **SHALL** remain secure until all keying material has been removed and/or destroyed.

## 7.4 Key Generation

Keys and key components **SHALL** be generated by using a random or pseudo-random process such that it is not feasible to determine that certain keys are more probable than other keys from the set of all possible keys. A variant of a key **SHALL** be used only for key separation and not for key generation. Key components **SHOULD** be generated with a check value. When a TRSM generates a key for transfer from the TRSM, a cryptographic check value **SHOULD** be generated.

When a check value is generated for a key or key component, it **SHALL** be generated by an encryption process (e.g., using the TECB mode of Reference 2), such that all portions of the key or key component are involved in generating the check value. It is a significant security risk to generate check values on portions of the key individually.

## 7.5 Symmetric Key Distribution

### 7.5.1 Manual Distribution

Manual distribution of an initial cryptographic key **SHALL** be performed in one of the following ways. For additional implementation details, see Annex C.

The first way is under dual control in two or more key components with split knowledge. Each of these key components **SHALL** be the same full length as the cryptographic key. Each entity (usually person or group of persons) responsible for each key component **SHALL** be instructed to keep secret the key component entrusted to them. Each key component **SHALL** be subject to the following:

1) If the key component is not in human comprehensible form (e.g., in a memory module, in a smart card, in a magnetic stripe card), it is in the physical possession of only one entity for the minimum practical time until the key component is entered into a TRSM; and

2) If the key component is in human comprehensible form (e.g., printed as within a PIN-mailer-type document) it is visible only at one point in time to only one person, and only for the duration of time required for this person to privately enter the key component into a TRSM; and

3) The key component is never in the physical possession of an entity when any one such entity is or ever has been similarly entrusted with any other component of this same key; and

4) The key component is entered directly into a TRSM in a secure environment.

The resultant cryptographic key **SHALL** exist only within the TRSM by automatically combining all entered key components using XOR.

Any other TRSM loaded with the same key components **SHALL** combine all entered key components using the identical process. To facilitate the detection of errors in key distribution, the check value for key components and keys **SHOULD** be verified. (See Annex C.)

A second way for manually distributing a cryptographic key involves transferring a clear key from a Key Loading Device (KLD) to a directly connected TRSM. This **SHALL** be performed under dual control by two or more trusted individuals. In this case the key is under dual control because the

TRSMs are under dual control. An example could be the use of a Key Loading Device. This differs from the preceding case where the key is under dual control because the key components are under dual control.

Several key management methods may require implementation of additional facilities, functions, or devices as part of the overall security process. Such security facilities and their requirements are outlined below.

### 7.5.2 Key Initialization Facility

A Key Initialization Facility (KIF) is a secure facility which contains a TRSM and provides the initial keys to transaction terminal TRSMs and/or the acquirer's TRSMs.

The KIF may directly load the initial key into the terminal's TRSM (e.g., a PIN Entry Device), if the latter is portable and may be feasibly transported to the KIF. If the terminal's TRSM is not portable, the initial terminal key and its identifier **SHALL** be conveyed from the KIF to the terminal in a Key Loading Device (KLD). In either case, the transfer **SHOULD** be by electronic or other automatic means.

A KIF and transfer processes **SHALL** be implemented within a secure environment, and **SHALL** be operated under dual control. A KIF may be operated by an acquirer, and may be the same facility that the acquirer uses for the cryptographic operations involved in transaction processing. For additional implementation details see Annex C.

### 7.5.3 Key Loading Device

A Key Loading Device (KLD) **SHALL** contain or be a TRSM. When terminals or terminal sub-assemblies cannot feasibly be transported to the KIF, a KLD is used to convey keys from a KIF to one or more terminal's TRSM.

The KLD is physically transported to the KIF where one or more keys (and associated key identifiers, if required) are transferred into the KLD. The KLD is then physically transported to one or more terminal TRSMs.

The KLD **SHALL** be designed so that only authorized personnel can utilize and enable it to output a key into another TRSM. Such personnel **SHALL** ensure that the output is not being monitored.

The KLD **SHALL** be under constant supervision by a trusted person, or else **SHALL** be stored in a secured manner (e. g., in a safe) such that no unauthorized person may have access to it. The KLD **SHALL** be safeguarded in accordance with section 7.1 item h). For additional implementation details, see Annex C.

## 7.6 Key Utilization

Any key used by a communicating pair **SHALL** be unique (other than by chance). A key **SHALL** be used for one purpose only. However, a variant of a key may be used for a purpose different from that of the original key. Controls **SHALL** be provided to prevent key misuse. A variant of a key **SHALL** exist only in a device that possesses or possessed the original key.

Any key resident in a transaction-originating terminal **SHALL** exist only in that device and those facilities which are authorized to receive and/or transmit encrypted data or authentication elements from or to that device. The same key may be used to either (but not both) encrypt data or create authentication elements. This key may be used in one or both directions between these nodes.

If a transaction-originating terminal interfaces with more than one acquirer, then the transaction-originating terminal TRSM **SHALL** have a completely different and unique key or set of keys for each acquirer. These different keys, or set of keys, **SHALL** be totally independent and **SHALL** not

be variants of one another. TRSMs **SHALL** be implemented to ensure that misuse of the device will not result in the exposure of keys. For instance, it **SHALL** not be possible to command the TRSM to decrypt a key and expose it by telling the unit that the encrypted key is encrypted data. To this end, the TRSM **SHALL** provide a means to ensure the separation of keys according to cryptographic function.

The TRSM **SHALL** be designed, implemented, and controlled to prevent or detect its misuse to determine, by exhaustive trial and error, data secured by the TRSM.

### 7.7 Key Replacement

1) A cryptographic key **SHALL** be replaced with a new key within the time deemed feasible to determine the current key by exhaustive attack.

2) Keys **SHALL** be changed when compromise is known or suspected.

3) The replacement for a compromised key **SHALL** not be related in any way to the compromised key.

### 7.8 Key Destruction and Archival

Except for archival purposes, when keys have been compromised, suspected of having been compromised, or discontinued, all forms of such keys **SHALL** be destroyed.

Paper-based keying materials **SHALL** be destroyed by crosscut shredding, burning or pulping. All residue should be reduced to pieces 5 mm or smaller. When material is burned, the residue should be reduced to white ash. Keying material stored on other media **SHALL** be destroyed so that it is impossible to recover by physical or electronic means.

Destruction of keys **SHALL** be accomplished under conditions of full accountability, with appropriate records retained for audit trail purposes.

Archived keys **SHALL** be stored in accordance with section 7.1.

### 7.9 Key Encryption/Decryption

While not described in this standard, any method of key encryption/decryption **SHALL** operate in accordance with section 7.1.

## 8 Key Management Specifications

### 8.1 General

Terminal key management poses unique problems because of the potentially very large number of ATM and POS terminals, and the need for a card acceptor to install, move, and replace such terminals without being hampered by undue logistical restraints. Among the more significant problems are:

— loading the key into the terminal,

— determining at the acquirer what keys are in any given terminal at a particular point in time,

— identifying both the key and key management method unambiguously.

This section specifies:

- methods of key management,
- key identification techniques, and
- the data elements necessary for the transfer of security or key management information.

Although this part of this standard specifies the keying relationships between the terminal and the first node, any of these techniques may be used for any other nodes.

## 8.2 Methods of Key Management

This part of this standard recognizes methods which **SHALL** be used singularly or in combination for key management. These methods fall into two broad categories, each of which may be appropriate to specific types of TRSMs.

a) Key Management Methods Requiring Compromise Prevention Controls

- Fixed Transaction Keys
- Master Keys / Transaction Keys

b) Key Management Method Requiring Compromise Detection Controls:

- Derived Unique Key Per Transaction (DUKPT)

The requirements for each method are specified in later sections but are summarized below.

### 8.2.1 Key Management Methods Requiring Compromise Prevention Controls

This category contains two methods of key management that require TRSMs that rely completely on physical barriers and security procedures. See also Section 7.2. Both methods use keys for more than one transaction. Both methods are suitable for use between any TRSMs at any nodes (including terminals).

#### 8.2.1.1 Fixed Transaction Keys

This method of key management uses keys for transaction processing that are distributed using some physical process, e.g., the device keypad, magnetic cards, key loading device. The keys are replaced by the same methods whenever compromise is known or suspected. This method is described in Section 8.5.

#### 8.2.1.2 Master Keys / Transaction Keys

This method uses a hierarchy of Key Encrypting Keys and Transaction Keys. The highest level of Key Encrypting Key is known as a Master Key. Master Keys are distributed using some physical process, e.g., the device keypad, magnetic cards, key loading device. Master Keys are replaced by the same methods whenever compromise is known or suspected.

Transaction Keys are distributed and replaced encrypted under a Key Encrypting Key. In a two-layer hierarchy, the Master Key is used to encrypt Transaction Keys directly. Alternatively, multiple levels of Key Encrypting Keys may be used. Each Key Encrypting Key is distributed and replaced encrypted under the next-higher level Key Encrypting Key.

This method is described in Section 8.6.

#### 8.2.2 Key Management Method Requiring Compromise Detection Controls

The second category contains a method to determine and generate keys that are unique for a given transaction. This method allows the use of TRSMs which do not rely completely on physical barriers and security procedures. This method is intended for use in applications where many transaction-originating TRSMs, which rely upon DUKPT to preclude the determination of past keys, communicate with a relatively few receiving TRSMs, which rely upon physical barriers to preclude the determination of past keys. This method is suitable for use with POS PIN Entry Devices, where many such PIN Entry Devices operate with the same acquirer(s). This method is suitable for use between the TRSM of the terminal and the TRSM at the first receiving node that cryptographically processes the transaction.

DUKPT relies on the use of a 'base derivation' key present only in the TRSM of the first receiving node that cryptographically processes that transaction. The unique Transaction Keys used by the TRSMs of terminals are derived from the Base Derivation Key using only non-secret data transmitted as part of each transaction.

This method is described in Section 8.7.

### 8.3 Key Identification Techniques

At the time of the transaction, both the terminal TRSM and the TRSM of the first receiving node to cryptographically process the transaction need information so as to determine the associated Transaction Key.

In certain circumstances, normal transaction data alone may be sufficient for the node to identify the key. In other circumstances, key management information is sent as part of the transaction message so the Transaction Key can be identified or derived. Selected transaction data elements may be combined to identify uniquely the key used in the transaction processing. The elements selected may depend on the capabilities of the terminal, those of the node, the key management methodology, and the key identification technique used.

One of the following techniques **SHALL** be used to determine a Transaction Key.

#### 8.3.1 Implicit Key Identification

Additional specific information for solely identifying the Transaction Key(s) is not contained in the transaction messages, rather information normally present in the transaction may be used. For example, the terminal identification number in a transaction message is used as a reference to a Transaction Key database to identify the key.

#### 8.3.2 Key Identification by Name

Additional specific information that names a Transaction Key is included in the transaction message. This key name can be used to reference or derive the Transaction Key. As an example, a reference key name (independent of the terminal identification number) is included in the message for the purpose of obtaining the Transaction Key from a database. As another example, a key name is included in the transaction message which, if cryptographically processed using the correct Derivation Key, yields the Transaction Key.

### 8.4 Security Management Information Data (SMID) Element

In order to implement key management, data elements may be necessary for the transfer of key management information. Specifically, a data element is needed to provide the framework to specify a methodology and to specify the key identification.

The Security Management Information Data Element (SMID) may be used to convey security or key management information for use in the current transaction message or future transaction messages.

1) The current transaction message

The SMID is typically used to identify the current Transaction Key. For example, the SMID may be used in a point-of-sale environment in which thousands of terminals interface with the same acquirer. In this case the SMID is transmitted from the terminal to the acquirer, and enables the acquirer to determine the key(s) used at the terminal for encryption/authentication in the associated message.

The SMID is sufficient to uniquely identify the key(s) used with the associated transaction message. When there is only a single recipient for each transaction message originator, and this recipient is not expected to change, the fields of the SMID may be specified by predefined agreement between the transaction originator and the transaction recipient. When there are, or may be multiple transaction recipients, or when the transaction recipient may change (e.g., the merger of financial institutions), it is recommended that the SMID be coded in a standardized manner. This is especially true if a given transaction message originator may interface with additional recipients in the future, to ensure that there is no ambiguity or inconsistency between the coding of the SMID in transactions which the recipient receives from other originators.

In certain implementations, the SMID may be omitted. In those situations, the key management method defaults to a method predetermined by the entities , and the key is implicitly identified. For example, a card acceptor may operate terminals which do not transmit a SMID, provided the card acceptor already knows which method was installed in the terminal, and can implicitly determine the current Transaction Key. Use of such defaults (i.e., implicitly specified) may limit future flexibility to accommodate the implementation or coexistence of other key management methods.

2) Future transaction messages

The SMID is alternatively used to convey security or key management information for future transaction messages. For example, when used between a terminal and an acquirer, the SMID could be sent from terminal to acquirer when identifying current keys, and from acquirer to terminal when conveying information about future keys. Another example is to convey security or key management information between two nodes that use the Master Key/Transaction Key method. Some key management methods never use the SMID to convey information about future keys.

A special feature of the SMID permits several SMIDs to be conveyed in a single message. This feature also allows information for either or both of the two purposes above.

The following subsections describe the SMID in terms of its structure, representation, fields, attributes, encoding and key naming conventions. In later sections of this part of this standard, each key management method specifies its data subfields and how they are utilized.

Even though each implementation of different key management methods may specify how the SMID is to be structured and used to convey keying information, the SMID, when present, **SHOULD** conform to the following specifications. When the SMID appears in a message in accordance with Reference 7, the SMID **SHALL** conform to the specifications in this section.

Reference 7 SMID field location:

The SMID is defined in Reference 7 as follows:

| | | |
|---|---|---|
| Bit 52 | b-8 | PIN data (binary fixed length 8 bytes) |

| | | | |
|---|---|---|---|
| Bit 53 | LLVAR | b...48 | Security Control Information (binary variable length max 48 bytes) |
| Bit 64 | | b-8 | MAC (binary fixed length 8 bytes) |
| Bit 96 | LLLVAR | b...999 | Key Management Data (binary variable length max 999 bytes) |
| Bit 128 | | b-8 | MAC (same as Bit 64) |

Bit 53 of the Bit Map specifies the presence of the SMID when used in conjunction with bit 52 for 11xx authorization and 12xx financial transactions containing a PIN.

Bit 53 of the Bit Map specifies the presence of the SMID when used in conjunction with bit 128 (or 64) when any transaction is MACed. Thus, if a 1200 financial message includes a PIN (bit 52) and is MACed (bit 128) then bit 53 includes the SMIDs for both the PIN Key and the MAC Key.

Bit 96 of the Bit Map specifies the presence of the SMID when used for 18xx network key management transactions. An 1804 network management request message for key exchange contains a SMID (indicated by bit 96) for future transactions (e.g., new PIN Key for 1200 financial messages) and if the 1804 is MACed (indicated by bit 128) the 1804 contains another SMID (indicated by bit 53) for the current transaction (i.e., the MAC).

See Annex B for some SMID examples.

### 8.4.1 Notations, Abbreviations and Conventions

Abbreviations used in this part of this standard to represent SMID and field attributes, format, and usage are identified in this section.

Representation Attributes:

| | |
|---|---|
| n | numeric digits |
| b | binary representation of data, where the length attribute represents the number of bytes present (8 bits to 1 byte) |

Length Attributes:

| | |
|---|---|
| 3 | fixed length of 3 bytes |
| 1...17 | variable length up to a maximum of 17 bytes with a minimum length of 1 byte. |

Note: The length at the beginning of the SMID or, if present, of a variable length field will identify the number of positions following to the end of the data element or field.

Length Format:

| | |
|---|---|
| LLL | length of a variable SMID |
| VAR | variable length SMID<br>For example, in a SMID represented as LLLVAR, there is a three digit prefix (LLL) that identifies the number of positions occupied by the data (VAR). When the data is 42 characters, LLL=042. |

Field Usage:

| | |
|---|---|
| Mandatory | Signifies that the field **SHALL** be present. |
| Optional | Signifies that the data element or the field is optional, and that common usage has been identified; support by the recipient of the optional field is by mutual agreement between sender and recipient. |

Conventions:

All fields present in a SMID appear from left to right in the order described in this document.

The field attributes, representations, and descriptions are documented in a manner similar to standard data element descriptions in Reference 7.

### 8.4.2 Representation

The SMID (Data Element 53 and/or 96) is structured as follows:

| LLL | SMID data fields |
|---|---|

SMID B 1... 999 Optional

LLLVAR

A series of values used to identify the current Transaction Key, or to transfer security or key management information for future use.

The SMID is an optional binary field containing 1 to 999 bytes of data.

The SMID contents are defined below.

Other SMID fields required for a specific key management method are defined in the applicable section for that method. Note that the method may specify additional requirements for the fields defined below.

LLL n 3 Mandatory

Specifies the length, in bytes, of the remainder of this data element. The LLL itself is coded in accordance with the applicable message standards and implementation specifications.

CONTROL B 1 Optional

A value in the range of A0 to FF inclusive, used to determine the purpose of the SMID and/or the key management method and/or interpretation of the subsequent SMID fields.

A value in the range of 00 to 9F, inclusive, denotes that CONTROL is omitted, and the SMID identifies the current Transaction Key. In this case, this byte is the first byte in the KEY NAME field which **SHALL** utilize the Key Set Identifier (see Section 8.4.3, Key Naming).

| Value | Meaning |
|---|---|
| A0 | Reserved for future ANSI use. |
| A1 ... AF | The SMID identifies key management information for use in the associated transaction message. The key management method (or other indicator) is specified by the value. Subsequent SMID fields used to identify the current Transaction Key are not denoted/delimited by length subfields. The result is concatenated fixed length fields of length specified by pre-determined agreement. |

| field 1 | . . . | field n-1 | field n |
|---|---|---|---|

where:

A1 Method: Fixed Transaction Keys (see Section 8.5)

A2 Method: Master Keys/Transaction Keys (Section 8.6)

A3 Reserved for future ANSI use

A4 Method: Single DEA DUKPT (no longer supported by this standard)

A5 Method: Triple DEA DUKPT (Section 8.7)

A6...AF Reserved for future ANSI use

B0 This SMID is employing the special feature permitting several SMIDs to be conveyed in a single message. Subsequent SMID fields identify the number and purpose of SMID fields and, except for the COUNT field, are denoted/delimited by length subfields as follows:

| COUNT | length | SMID 1 | . . . | length | SMID n-1 | length | SMID n |
|---|---|---|---|---|---|---|---|

COUNT B 1 Mandatory

A binary value (0-9) indicating the count of SMID fields to follow.

length B 1 Mandatory

A binary value (1-255) which specifies the length, in bytes, of the remainder of the 'SMID i'. Note that a zero value (0) denotes a missing field.

SMID i B 1... 255 Optional

A SMID, as defined in this standard (exclusive of the length field LLL), e.g., CONTROL, KEY NAME, etc. The purpose of each SMID is determined independently.

To minimize complexity and confusion when using multiple SMIDs, each 'SMID i' SHOULD not specify a CONTROL value 'B0'.

B1...BF The SMID is identifying key management information for use in the associated transaction message. The key management method (or other indicator) is specified by the value. Subsequent SMID fields used to identify the current Transaction Key are denoted/delimited by length subfields as follows:

| length | field 1 | . . . | length | field n-1 | length | field n |
|---|---|---|---|---|---|---|

length B 1 Mandatory

A binary value (1-255) which specifies the length, in bytes, of the remainder of 'field i'. Note a zero value (0) denotes a missing field.

where:

B1 Method: Fixed Transaction Keys

B2 Method: Master Keys/Transaction Keys

B3 Reserved for future ANSI use.

B4 Method: Single DEA DUKPT (no longer supported by this standard)

B5 Method: Triple DEA DUKPT

B6...BF Reserved for future ANSI use.

C0 Reserved for future ANSI use.

C1...CF The SMID is conveying security or key management information to be used in future transaction messages. The key management method (or other indicator) is specified by the value. Subsequent SMID fields used to convey specific key management information are specified in the method sections of this standard.

where:

C1 Reserved for future ANSI use

C2 Method: Master Keys/Transaction Keys

C3 Reserved for future ANSI use

C4 Reserved for future ANSI use

C5...CF Reserved for future ANSI use.

D0...DF Reserved for private/proprietary use.

E0...EF Reserved for future ANSI use.

F0...FF Reserved for future ANSI use.

KEY NAME B1...16 Optional

A value used to identify, name, index, derive or determine a Transaction Key. A specific key management method may impose further requirements on the KEY NAME such as

- defining how it is used
- defining how it is formatted/encoded
- restricting its size range

If the CONTROL field is omitted from the SMID, then the KEY NAME **SHALL** utilize the Key Set Identifier (see Section 8.4.3, Key Naming).

Note that formatting/encoding requirements for different key management methods may affect the encoding or interpretation of a specific key name or recognition of a CONTROL field. For example, Master Keys/Transaction Keys may require KEY NAME as coded characters.

### 8.4.3 Key Naming

Key Names **SHOULD** be represented according to the following conventions:

1) The left-most portion of the Key Name (e.g., KEY NAME) is a Key Set Identifier. This Key Set Identifier, described below, **SHOULD** be assigned in accordance with Annex E.

2) The right-most portion of the Key Name may contain one or more subfields. These subfields may be formatted or utilized in a unique manner as required by the specific key management method or implementation. Taken together, such subfields uniquely identify a key within a key set and, when concatenated with the associated Key Set Identifier, form the unique Key Name.

All keys associated with transaction message originators may be divided into 'sets' and denoted by Key Set Identifiers. A Key Set Identifier **SHALL** start with a value in the range 00-9F. Although all keys within a 'set' are unique, and the compromise of one such key need not disclose any other key of the set, all keys in the same set have certain common characteristics. These are:

— Use of the same base derivation key. If a recipient determines Transaction Keys by a decryption, transformation, or derivation process, all keys within a given key set use the same Key Encrypting Key, or use the same transformation or Derivation Key to determine the Transaction Key. (Note that in a POS environment, such 'global' keys may be known only at the acquirer and not at terminals.)

— A common structure to the remainder of the SMID. This makes it possible to interpret the Key Identifier subfield without the need for delimiters or embedded 'length' bytes to locate the boundaries between any other subfields which this Key Identifier may contain.

— Use of the exact same key determination methodology. All keys within a key set are determined using the identical software or firmware process. Furthermore such software or firmware is able to determine any Transaction Key from this key set given only the base derivation key as determined by the key set identifier and the SMID for this transaction.

— Identically formatted key-related SMID, if required, returned to the transaction message originator. If the key-related information is to be returned to the transaction message originator with the transaction response message, such information is identically formatted for all of those transaction message originators whose keys are in the same key set.

If transactions from one transaction message originator may go to multiple recipients, but the originator cannot determine which recipient will receive any given transaction, then all of the recipients **SHALL** share the base derivation key associated with the corresponding key set, and all **SHALL** implement the appropriate key determination methodology.

## 8.5 Method: Fixed Transaction Keys

In this method the physically loaded key(s) **SHALL** be used for transaction processing until such time as a new key(s) is physically loaded.

### 8.5.1 SMID

If a SMID is used with this method, the Transaction Key is identified (see Section 8.4) as follows:

| | |
|---|---|
| CONTROL | Optional<br>Value is hex A1 or B1 |
| KEY NAME | Mandatory<br>The name, nickname, abbreviation or indicator of the Transaction Key. Such naming **SHALL** be established by predefined agreement between the two entities. |

#### 8.5.2 Additional Key Management Requirements

##### 8.5.2.1 Initialization

Key initialization **SHALL** be performed in accordance with Section 7.5 Symmetric Key Distribution.

##### 8.5.2.2 Cryptographic Key Synchronization

Cryptographic key synchronization **SHOULD** be confirmed prior to key usage by the communicating pair.

Recovery from cryptographic key synchronization loss requires re-initialization.

#### 8.5.3 Additional Notes

1) This method is not associated with the need for online key management as all keys are physically loaded (see Section 7.5). Regular manual handling of keying material exposes the system to potential compromise. Therefore this method will normally involve keys with an extended life. If this method is used, keys **SHALL** be double-length.

2) When the SMID is present in the message, it is subject to alteration. In developing their predefined agreement, the two entities **SHOULD** analyze the risks and consequences of alteration as it relates to the specific implementation. It **SHOULD** be noted that some implementations may not suffer adversely. If it is determined that alteration is a concern, then the SMID **SHOULD** be included as an authentication element (see Reference 5).

3) A predefined agreement establishing the naming and character code of KEY NAME may be needed since the SMID is a binary data element.

4) This method is intended to be simple. Since no CONTROL 'C1' is specified, if any additional specific key management information (e.g., initialization vectors) is needed, it **SHALL** be specified and implemented by predefined agreement or by using another appropriate key management method.

5) This method **SHALL** be used only in a TRSM which relies exclusively on physical barriers (see Section 7.2).

### 8.6 Method: Master Keys / Transaction Keys

This method uses a hierarchy of keys. At the lowest level of this hierarchy are Transaction Keys used for transaction processing; these are also known as Data Keys. Transaction Keys are distributed and replaced under a Key Encrypting Key. Each Key Encrypting Key is encrypted under the next higher level Key Encrypting Key. The highest level Key Encrypting Key is the Master Key, which **SHALL** be physically loaded.

To utilize this method, the basic mechanisms regarding all aspects of key management, including Cryptographic Service Messages (CSMs), **SHOULD** be implemented. (See Annex D for information on CSMs.)

#### 8.6.1 SMID

##### 8.6.1.1 Key Identification

If a SMID is used with this method, the Transaction Key is identified (see Section 8.4) as follows:

| | |
|---|---|
| CONTROL | Optional<br>**Value is hex A2 or B2.** |
| KEY NAME | Mandatory<br>The name or key identifier of the key transmitted. |

#### 8.6.1.2 Key Replacement

Key replacement **SHALL** be accomplished using CONTROL to convey a CSM in the SMID. Any valid CSM may be presented; however, it is not a requirement of this standard that every CSM be implemented.

For this method, specific key management information is conveyed as follows:

| | |
|---|---|
| CONTROL | Mandatory<br>**Value is hex C2.** |
| CSM | B1...998 Mandatory<br>Since the CSM is the remainder of the SMID, no length subfield is specified. |

### 8.6.2 Additional Key Management Requirements

#### 8.6.2.1 Initialization

Key initialization **SHALL** be performed in accordance with Section 7.5 Symmetric Key Distribution.

#### 8.6.2.2 Cryptographic Key Synchronization

Cryptographic key synchronization **SHOULD** be confirmed prior to key usage by the communicating pair.

Recovery from cryptographic key synchronization loss requires re-initialization or transmission of a new Transaction Key.

### 8.6.3 Additional Notes

1) When the SMID is present in the message, it is subject to alteration. When the SMID is being used to convey a future key, the SMID contains a CSM. Since the CSM has its own message authentication code, it is not a requirement to authenticate a network management message, for example, containing only a CSM. However, if such a CSM is contained in a SMID within a retail financial transaction message, the CSM will have its own MAC and the transaction message may have another MAC computed in accordance with Reference5.

2) A predefined agreement establishing the naming and character code of KEY NAME may be needed since the SMID is a binary data element. CSMs supplied within a SMID will also be treated as binary data. Thus these coded-character messages will not be translated from character-set to character-set as they traverse a network. For certain implementations, this may be desirable due to increased efficiency; other implementations may be impacted.

3) The CSMs **SHOULD** be independent of the messages or message types used to convey them. See Reference 7.

4) Multiple CSMs may be combined in a block using the CONTROL 'B0' option. Also note that the 'SMID i' subfields have a length limit which may impact a few CSMs.

5) A very brief summary of the hierarchy and automated distribution of keys is presented in Annex D for information only.

6) This method **SHALL** be used only in a TRSM which relies exclusively on physical barriers (see Section 7.2).

7) Since the Master Key is a Key Encrypting Key, it **SHALL** be at least double length.

## 8.7 Method: DUKPT (Derived Unique Key Per Transaction)

With this method each transaction-originating TRSM uses a unique key for each transaction, yet never contains any information which would allow the determination of any key previously used by this TRSM, nor of any key which has been or will be used by any other transaction-originating TRSM. The receiving TRSM **SHALL** determine the current Transaction Key used by any transaction-originating TRSM from:

1) the non-secret information contained in the transaction's SMID, and

2) a Base Derivation Key. This Base Derivation Key:

   — **SHALL** reside in a TRSM which relies exclusively on physical barriers

   — resides in one or more receiving (e.g., acquirer's) TRSMs

   — does not reside in any originating (e.g., terminal's) TRSM

   — is used to generate the originating TRSM's unique double-length Initial Key using the KEY NAME

   — can be used to generate the unique double-length Initial Keys for many originating TRSMs

   — **SHALL** be a double-length or triple-length key.

This method **SHALL** operate at the receiving TRSM, as shown in Figure 1. The SMID **SHALL** consist of a KEY NAME which **SHALL** utilize a KEY SET IDENTIFIER. The KEY NAME consists of three subfields. The left-most subfield is a KEY SET IDENTIFIER which is used to select the Base Derivation Key appropriate to the TRSM originating the transaction. The second subfield is a TRSM IDENTIFIER, and the concatenation of the KEY SET IDENTIFIER and the TRSM IDENTIFIER is encrypted using the selected Base Derivation Key. The result is the Initial Key which had been loaded into the originating TRSM (though erased from this TRSM, perhaps years ago). The third subfield is a TRANSACTION COUNTER. The originating TRSM **SHALL** increase its TRANSACTION COUNTER for each transaction. The originating TRSM **SHALL** cease operation when its TRANSACTION COUNTER overflows to zero. The receiving TRSM **SHOULD** verify that the originator's TRSM TRANSACTION COUNTER in the SMID has increased.

The Initial Key and the TRANSACTION COUNTER are inputs to a non-reversible transformation process which produces the Transaction Key used for the current transaction. In the example implementation (see Annex A) of this method, the transformation process requires no more than 10 TDEA cycles even though the TRANSACTION COUNTER can have more than a million different values.

Note that the initially loaded key is a function of the Base Derivation Key, the KEY SET IDENTIFIER, and the TRSM IDENTIFIER. Therefore no two originating TRSMs will be given the

same Initial Key provided that no two originating TRSMs with the same KEY SET IDENTIFIER have identical TRSM IDENTIFIERs.

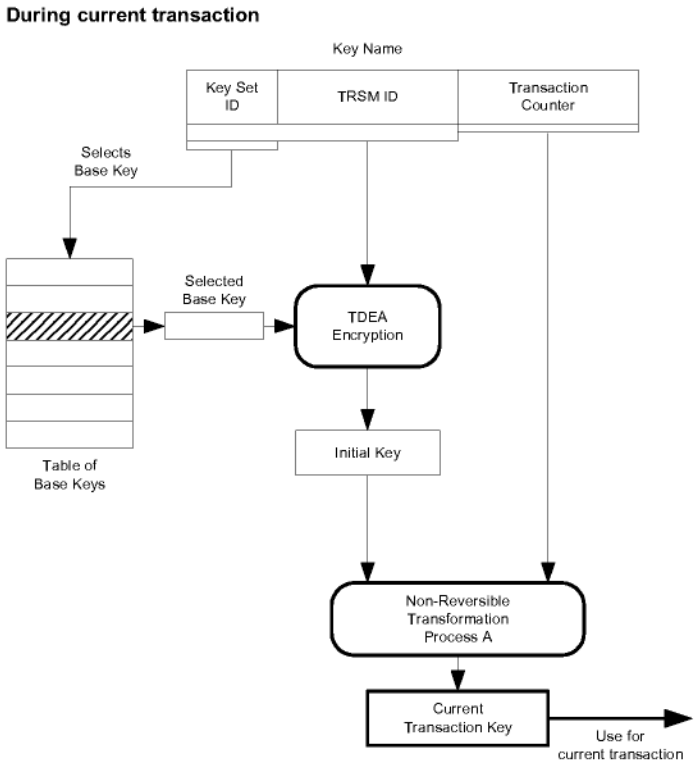



**Figure 1 – DUKPT at Receiving TRSM**

The originating TRSM **SHALL** generate and use a current Transaction Key such that the receiving TRSM can determine that key using the process shown in Figure 1. The originating TRSM **SHALL** also erase all record of the current Transaction Key immediately after completion of the current transaction.

An example of how this method operates at the originating TRSM is shown in Figure 2. This TRSM stores a number of future keys. At the beginning of a new transaction, the TRANSACTION COUNTER (the right-most portion of the KEY NAME ) is incremented, and then is used to select one of these future keys as the current Transaction Key. The selected key is erased from future key storage. Note that the KEY NAME is transmitted in the SMID with the current transaction.

At the completion of the transaction, some number of future keys (sometimes none, sometimes one or more) are generated by non-reversibly transforming the current Transaction Key as a function of the TRANSACTION COUNTER. These newly generated future keys are then stored into those locations in future key storage determined by the TRANSACTION COUNTER. The current Transaction Key is then erased. Therefore, the TRSM retains no information about any key used for any previous transaction.

In Figure 1 and Figure 2, the non-reversible transformation processes 'A' and 'B' are different but related. Future keys are generated, stored, and selected at the originating TRSM in a manner such that the receiving TRSM is able to determine the current Transaction Key.

Annex A specifies the implementation of this technique.

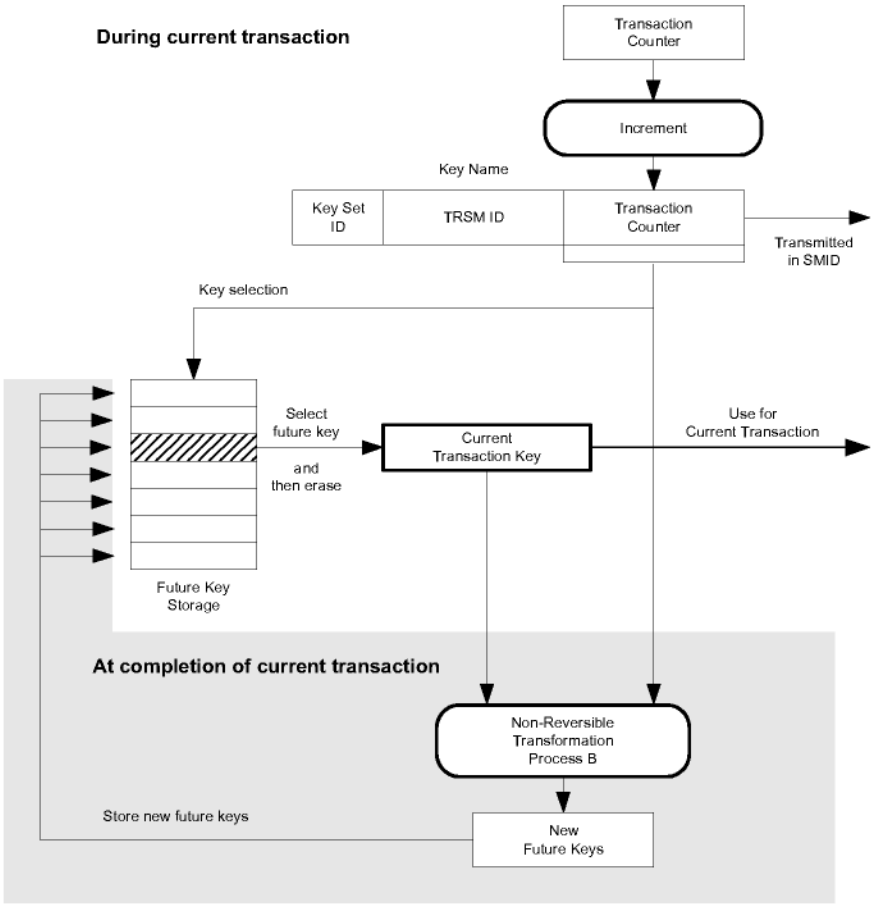



**Figure 2 – DUKPT at Originating TRSM**

### 8.7.1 SMID

For this method, the Transaction Key is identified (see Section 8.4) as follows:

| | |
|---|---|
| CONTROL | Optional<br>Value is hex A5 or B5. |
| KEY NAME | B 5...16 Mandatory<br>The value used to derive the current Transaction Key. The KEY NAME **SHALL** incorporate a KEY SET IDENTIFIER. |

### 8.7.2 Additional Key Management Requirements

#### 8.7.2.1 Initialization

Both the Initial Key and the associated KEY NAME **SHALL** be loaded into the originating TRSM. The originating TRSM **SHALL** ensure that the TRANSACTION COUNTER is set to zero. The Base Derivation Key used to derive this Initial Key **SHALL** be available to the receiving TRSM before it can cryptographically process data from the originating TRSM. It **SHALL** be ensured that no two originating TRSMs initialized with the same KEY SET IDENTIFIER are initialized with the same TRSM IDENTIFIER.

#### 8.7.2.2 Cryptographic Key Synchronization

The method is inherently self-synchronizing.

Special recovery is not required because synchronization cannot be lost.

### 8.7.3 Additional Notes

1) Other system, communication, or operational failures may appear to impact cryptographic key synchronization.

2) This method does not require an acquirer to maintain any database of key management related data, except for a relatively small number of Base Derivation Keys which can be stored within the TRSMs of this acquirer.

3) Because in this method a transaction identifies its own Transaction Key, it is susceptible to a fraud threat: An adversary compromises one transaction-originating TRSM and then replicates the key identifier, in this case the KEY NAME (which forms the SMID transmitted by this TRSM), and the key itself, in many originating TRSMs. In this case this threat may be countered by two means: (1) The originating TRSM automatically sets its TRANSACTION COUNTER to zero when a new key is loaded. Since the compromised TRSM had a non-zero TRANSACTION COUNTER (the counter is incremented immediately after key loading), the fraudulently-loaded TRSM cannot be made to simulate the compromised TRSM. (2) An audit process is performed, perhaps off-line using logged KEY NAMEs, to determine if two or more transaction-originating TRSMs are associated with the same KEY SET IDENTIFIER and TRSM IDENTIFIER. This audit process detects the use of a counterfeit TRSM which has been loaded with data from a compromised TRSM.

4) This method allows transaction-originating TRSMs to be loaded with Initial Keys and initial KEY NAMEs prior to being distributed to their operational locations. If the TRSMs are PIN Entry Devices which are used with POS terminals, this means that the PIN Entry Devices may be delivered to the merchant's non-secure facility with the keys already loaded. This prevents an adversary from replacing a legitimate TRSM with a counterfeit unit which has all of the correct operational characteristics plus a secret data disclosing 'bug'. Assuming that the key is automatically erased if the TRSM is opened, it also prevents the placing of a secret data disclosing 'bug' within a legitimate TRSM. The counterfeit TRSM could not contain a valid key and therefore transactions from that TRSM would be rejected.

5) This method allows a TRSM which has been loaded with a key to be subsequently associated with a transaction-originating terminal without the need for any coordination with the acquirers. If the TRSMs are POS PIN Entry Devices, the merchant can install any PIN Entry Device at any time with any terminal and can replace a failed PIN Entry Device at any time with a spare unit, all without the need for manual coordination with the acquirers.

6) This method can be used with a 'sub-key' methodology which enables a set of TRSMs, for example a merchant's POS PIN Entry Devices, to change from one acquirer to another without the need to transfer to this second acquirer the Base Derivation Key which the first acquirer might use for many other merchants as well.

# Annex A
# (Informative)
# Derived Unique Key Per Transaction

(This Annex is not part of this standard and is included for information only.)

This annex describes the TDEA Derived Unique Key Per Transaction key management method for PIN Entry Devices. The technique as described supports double-length TDEA keys and not triple-length TDEA keys. In addition to deriving a unique PIN Encryption key for each transaction, the method includes the optional derivation of unique-per-transaction MAC keys and Data Encryption keys.

Initially, the annex describes the various preferred PIN Entry Device storage areas and then specifies the preferred PIN Entry Device functions that are used for the triple DEA method of derived unique key per transaction of processing PINs. Either the methodology as described below or its functional equivalent is performed to ensure that the key serial number and encrypted PIN block are generated correctly.

In any of the descriptions below, the bit or byte order is assumed to be such that the left-most bit, decimal digit, hexadecimal digit, or byte is the most significant and the right-most bit, decimal digit, hexadecimal digit or byte is assumed to be the least significant. In the case of the Shift Register, this implies that bit #1 is the most significant and bit #21 is the least significant.

## A.1 Storage Areas

The PIN Entry Device maintains certain storage areas only during the PIN processing operation. Other storage areas are permanently maintained.

### A.1.1 PIN Processing

The contents of the following storage area relating to PIN processing is maintained only during a given PIN encryption operation:

**Account Number Register (12 decimal digits)**
Holds the 12 right-most digits (excluding the check digit) of the primary account number received from the terminal in the "Request PIN Entry" command.

### A.1.2 Key Management

The following storage areas relating to key management are maintained from the time of the "Load Initial Key" command for the life of the PIN Entry Device:

**Initial Key Serial Number Register (59 bits)**
Holds the left-most 59 bits of the key serial number initially injected into the PIN Entry Device along with the initial PIN encryption key during the "Load Initial Key" command. The contents of this register remain fixed for the service-life of the PIN Entry Device or until another "Load Initial Key" command.

**Encryption Counter (21 bits)**
A counter of the number of PIN encryptions that have occurred since the PIN Entry Device was first initialized. Certain counter values are skipped (as explained below), so that over 1 million PIN encryption operations are possible. Note: The concatenation (left to right) of the Initial Key Serial

Number Register and the Encryption Counter form the 80-bit (20 hexadecimal digits) Key Serial Number Register.

**Future Key Registers (21 registers of 34 hexadecimal digits each)**
A set of 21 registers, numbered #1 to #21, used to store future PIN encryption keys. Each register includes a 2 hexadecimal digit longitudinal redundancy check (LRC) or a 2 hexadecimal digit cyclical redundancy check (CRC).

The following storage areas relating to key management are required on a temporary basis and may be used for other purposes by other PIN processing routines:

**Current Key Pointer (approximately 4 hexadecimal digits)**
Contains the address of that Future Key Register whose contents are being used in the current cryptographic operation.
**![Current Key Pointer]** identifies the contents of that Future Key Register whose address is contained in the Current Key Pointer.

**Shift Register (21 bits)**
A 21-bit register, whose bits are numbered left to right as #1 to #21. This register normally contains 20 "zero" bits and a single "one" bit. One use of this register is to select one of the Future Key Registers. The Future Key Register to be selected is the one numbered identically to the bit in the Shift Register containing the single "one".

**Crypto Register-1 (16 hexadecimal digits)**
A register used in performing cryptographic operations.

**Crypto Register-2 (16 hexadecimal digits)**
A second register used in performing cryptographic operations.

**Key Register (32 hexadecimal digits)**
A register used to hold a cryptographic key.

**MAC Key Register (32 hexadecimal digits)**
An optional register used to hold a cryptographic key for performing message authentication.

**MAC Response Key Register (32 hexadecimal digits)**
An optional register used to hold a cryptographic key for performing message authentication.

**Data Encryption Key Register (32 hexadecimal digits)**
An optional register used to hold a cryptographic key for performing data encryption.

**Data Encryption Response Key Register (32 hexadecimal digits)**
An optional register used to hold a cryptographic key for performing data encryption.

## A.2 Processing Algorithms

The PIN Entry Device may receive any of three different commands from the device to which it is attached. These commands are: "Load Initial Key", "Request PIN Entry", and "Cancel PIN Entry". The processing steps to be performed in each case are indicated below. Note: Whenever the text indicates that the LRC is used, either an LRC or a CRC may be used.

Subsequent to the following routines are definitions of subroutines that are common to them:

**"Load Initial Key" (External Command)**

1) Store the initial PIN encryption key, as received in the externally initiated command, into Future Key Register #21.

2) Generate and store the LRC on this Future Key Register.

3) Write the address of Future Key Register #21 into the Current Key Pointer.

4) Store the Key Serial Number, as received in the externally initiated command, into the Key Serial Number Register. (This register is the concatenation of the Initial Key Serial Number Register and the Encryption Counter.)

5) Clear the Encryption Counter (the 21 right-most bits of the Key Serial Number Register).

6) Set bit #1 (the left-most bit) of the Shift Register to "one", setting all of the other bits to "zero".

7) Go to "New Key-3".

**"New Key" (Local Label)**

1) Count the number of "one" bits in the 21-bit Encryption Counter. If this number is less than 10, go to "New Key-1".

2) Erase the key at ![Current Key Pointer].

3) Set the LRC for ![Current Key Pointer] to an invalid value (e.g., increment the LRC by one).

4) Add the Shift Register to the Encryption Counter. (This procedure skips those counter values that would have more than 10 "one" bits.)

5) Go to "New Key-2".

**"New Key-1" (Local Label)**

1) Shift the Shift Register right one bit (end-off). (A "zero" is shifted into position #1, the left-most bit of the register.)

2) If the Shift Register now contains all zeros (i.e., the single "one" was shifted off), go to "New Key-4", else go to “New Key-3”.

**"New Key-3" (Local Label)**

1) The Shift Register, right justified in 64 bits, padded to the left with zeros, OR'ed with the 64 right-most bits of the Key Serial Number Register, is transferred into Crypto Register-1.

2) Copy ![Current Key Pointer] into the Key Register.

3) Call the subroutine “Non-reversible Key Generation Process”.

4) Store the contents of Crypto Register-1 into the left half of the Future Key Register indicated by the position of the single "one" bit in the Shift Register.

5) Store the contents of Crypto Register-2 into the right half of the Future Key Register indicated by the position of the single "one" bit in the Shift Register.

6) Generate and store the LRC on this Future Key Register.

7) Go to "New Key-1".

**"New Key-4" (Local Label)**

1) Erase the key at ![Current Key Pointer].

2) Set the LRC for ![Current Key Pointer] to an invalid value (e.g., increment the LRC by one).

3) Add one to the Encryption Counter.

4) Go to “New Key-2”.

**"New Key-2" (Local Label)**

1) If the Encryption Counter contains all zeros, cease operation. (The PIN Entry Device is now inoperative, having encrypted more than 1 million PINs.) If not all zeros, go to “Exit”.

**"Exit" (Local Label)**

1) Return to original calling routine. (Processing of the current externally initiated command is completed, and the PIN Entry Device is ready for the next command. The Current Key Pointer, Account Number Register, Shift Register, and Crypto Pointer may now be used for other purposes.)

**"Request PIN Entry" (External Command)**

1) Transfer the primary account number as received in the externally initiated command into the Account Number Register.

2) Activate the PIN Entry Device keyboard and the Enter key.

3) If the PIN is not entered, send the encrypted PIN block response message without the PIN-related data elements and go to "Exit".

4) If the PIN is entered, use the cardholder-entered PIN and the primary account number to generate the cleartext PIN block and store it in Crypto Register-1.

5) Go to "Request PIN Entry 1".

**"Request PIN Entry 1" (Local Label)**

1) Call the subroutine "Set Bit".

2) Write into Current Key Pointer the address of that Future Key Register indicated by the position of the "one" bit in the Shift Register.

3) Check the LRC on ![Current Key Pointer]. If this byte is correct (valid key), go to "Request PIN Entry 2".

4) If the byte is incorrect, add the Shift Register to the Encryption Counter (to skip over the invalid key).

5) If the Encryption Counter contains all zeros, cease operation. (The PIN Entry Device is now inoperative, having encrypted more than 1 million PINs.)

6) Go to "Request PIN Entry 1".

**"Request PIN Entry 2" (Local Label)**

1) Copy ![Current Key Pointer] into the Key Register.

2) (Optional: Perform this step if you need to generate a key that will be used in a message authentication process; this step does not affect the generation of the PIN encryption key) XOR the value in the Key Register with hexadecimal “0000 0000 0000 FF00 0000 0000 0000 FF00” and save this resultant key in the MAC key register. If a separate key is used to verify the MAC response, XOR the value in the Key Register with hexadecimal “0000 0000 FF00 0000 0000 0000 FF00 0000” and save this resultant key in the MAC Response key register.

3) (Optional: Perform this step if you need to generate a key that will be used in a data encryption process; this step does not affect the generation of the PIN encryption key) XOR the value in the Key Register with hexadecimal "0000 0000 00FF 0000 0000 0000 00FF 0000". The resultant key is encrypted using itself as the key – see the One Way Function in Figure A-2 in section A.4.1. Save this resultant key in the Data Encrypting key register. If a separate key is used to encrypt the transaction response, XOR the value in the Key Register with hexadecimal "0000 00FF 0000 0000 0000 00FF 0000 0000". The resultant key is encrypted using itself as the key – see Figure A-2 in section A.4.1. Save this resultant key in the Data Encrypting Response key register. The input values into the One Way Function are not parity adjusted before they are used in the One Way Function.

4) XOR the Key Register with hexadecimal "0000 0000 0000 00FF 0000 0000 0000 00FF". (This will produce a variant of the key.)

5) Call the subroutine "Triple-DEA Encrypt".

6) Format and transmit the encrypted PIN block response message, which includes:

— The data in the Key Serial Number Register with leading hexadecimal "F's" suppressed (includes the 21-bit Encryption Counter).

— The encrypted PIN block in Crypto Register-1.

7) Go to "New Key".

Note: See Table A-1 in section A.4.1 which is a summary of the variants that may be used to modify the current key register to create separate keys for separate functions. The Data Encryption variant is the only variant that includes the use of a One Way Function to generate that separate key.

**"Cancel PIN Entry" (External Command)**

1) Deactivate the PIN Entry Device keyboard.

2) Go to "Exit"

The following routine may be used if the PIN Entry Device is implemented using electrically erasable programmable read-only memory (EEPROM). In this case, those storage areas that are most frequently rewritten (e.g., the four or so highest-numbered Future Key Registers and the corresponding bits of the encryption counter) are stored in volatile random access memory (RAM). Even though a power interruption results in the loss of the contents of the registers in volatile RAM, this loss is not significant, (if power to the PIN Entry Device were turned off and on again once a day for 10 years, the number of PINs the device could encrypt would be reduced by less than 5% due to "lost" keys).

Note: There may be a loss of synchronization if multiple power on reset cycles are executed without performing any transactions.

**"Power On Reset" (External Command)**

1) Set to "one" those bits of the Encryption Counter that correspond to the Future Key Registers lost because of the power interruption.

2) Increment the Encryption Counter.

3) Go to "Exit"

The following are subroutines used in the above processing routines:

**"Set Bit" (Local Subroutine)**

1) Set to "one" that bit in the Shift Register that corresponds to the right-most "one" bit in the Encryption Counter, making all other bits in the Shift Register equal zero. Example:

   — If Encryption Counter = 0 0010 1100 1101 1100 0011,
   Shift Register becomes: 0 0000 0000 0000 0000 0001;

   — If Encryption Counter = 0 0010 1100 1101 1100 0000,
   Shift Register becomes: 0 0000 0000 0000 0100 0000.

2) Return from subroutine to local calling routine.

**"Non-reversible Key Generation Process" (Local Subroutine)**

1) Crypto Register-1 XORed with the right half of the Key Register goes to Crypto Register-2.

2) Crypto Register-2 DEA-encrypted using, as the key, the left half of the Key Register goes to Crypto Register-2.

3) Crypto Register-2 XORed with the right half of the Key Register goes to Crypto Register-2.

4) XOR the Key Register with hexadecimal C0C0 C0C0 0000 0000 C0C0 C0C0 0000 0000.

5) Crypto Register-1 XORed with the right half of the Key Register goes to Crypto Register-1.

6) Crypto Register-1 DEA-encrypted using, as the key, the left half of the Key Register goes to Crypto Register-1.

7) Crypto Register-1 XORed with the right half of the Key Register goes to Crypto Register-1.

8) Return from subroutine.

**"Triple-DEA Encrypt" (Local Subroutine)**

1) Crypto Register-1 DEA-encrypted using the left half of the Key Register as the key goes to Crypto Register-1.

2) Crypto Register-1 DEA-decrypted using the right half of the Key Register as the key goes to Crypto Register-1.

3) Crypto Register-1 DEA-encrypted using the left half of the Key Register as the key goes to Crypto Register-1.

4) Return from subroutine.

## A.3 Key Management Technique

The specified key management technique provides a unique key per transaction with no information about any previous key retained in the PIN Entry Device. It also provides for a derived key system in which the Acquirer's security module can determine any key used at any time in any PIN Entry Device based on information included in the transaction.

For simplicity, assume that the Acquirer's security module also serves as the key loading device and injects the initial PIN encryption key and initial key serial number into each of the Acquirer's PIN Entry Devices. Contained within this highly secure security module is a derivation key. The module also contains a counter, which it increments each time a new PIN Entry Device is to be injected. To load a PIN Entry Device, the module encrypts this counter using its internally stored derivation key and transfers the result into the PIN Entry Device as the initial PIN encryption key. It also transfers the contents of the counter into the PIN Entry Device as the PIN Entry Device's

initial key serial number. In this way, every PIN Entry Device receives a unique initial PIN encryption key because each receives a unique initial key serial number.

The initial key serial number will remain unchanged for the operational life of the PIN Entry Device and will be included with every encrypted PIN block that the PIN Entry Device outputs. Thus, at any future time, when the security module receives a PIN block encrypted by this PIN Entry Device, the module can determine the initial PIN encryption key that it injected into the PIN Entry Device. (It simply uses its internally stored derivation key to encrypt the initial key serial number received with the encrypted PIN block.)

If the PIN encryption key remained constant for the operational life of the PIN Entry Device, it would be the initial PIN encryption key, the transmitted key serial number would consist of only the initial key serial number, and the above would be the extent of the key-related operations. However, as previously indicated, the key changes on every encryption, and no meaningful residue of any past key may remain in the PIN Entry Device.

To describe how this is accomplished, within the context of a derived key system, we will first describe a simplified technique that is not the one actually implemented but that is easy to understand and can in turn be used to explain the actually implemented system.

This simplified system operates as in the following diagram. When the initial PIN encryption key is injected into the PIN Entry Device, it is concatenated to the right with an encryption counter, which is initialized to zero. This counter increments after each PIN encryption. Each time the encryption counter increments, it is encrypted using the old PIN encryption key, and the resulting cipher becomes the new PIN encryption key. Thus, the new key is a "nonreversible transformation" of the old key. Given knowledge of the new PIN encryption key, there is no feasible way to determine the old PIN encryption key.

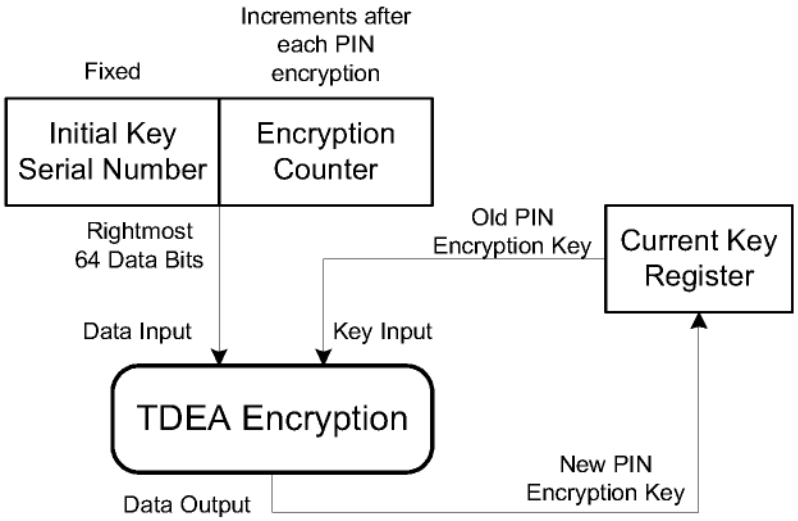



**Figure A-1 – Simplified DUKPT Data Flow**

In every encrypted PIN block response message, the PIN Entry Device transmits both the encrypted PIN block itself and also the concatenation of the initial key serial number and the encryption counter. The Acquirer is able to determine the initial PIN encryption key (by encrypting the initial key serial number using the derivation key stored in the security module). Having done this, the Acquirer is then able to perform the process indicated in the above diagram with a counter value (binary) of ...000001, then repeat this process for a counter value of ...000010, then ...000011, etc., up to the value of the encryption counter included in the encrypted PIN block message. The result of these multiple passes through the above process is the current PIN encryption key used to encrypt the PIN block.

The above procedure, though theoretically possible, is clearly not feasible. The expected life of a PIN Entry Device is up to 1 million PIN encryptions, so an excessive number of encryption cycles would be required by the security module to derive the current PIN encryption key once the encryption counter had reached a reasonably high count.

What is needed is a "key transformation" technique that the security module can follow without going through all of the intervening keys. That is, the module is able to take "giant steps" forward and not regenerate every single intervening PIN encryption key to reach the current one.

There are several ways in which this objective can be accomplished. The method chosen here is to make each PIN encryption key a nonreversible transformation not (necessarily) of the immediately preceding key but rather of that PIN encryption key for which the encryption counter contained the same bit configuration less the right-most "one" bit. In other words:

| The PIN encryption key corresponding to encryption counter value: | Is a nonreversible transformation of the PIN encryption key corresponding to encryption counter value: |
|---|---|
| 0000 1011 0011 | 0000 1011 0010 |
| 0000 1011 0010 | 0000 1011 0000 |
| 0011 0000 1000 | 0011 0000 0000 |
| etc. | etc. |

As a result, a number of PIN encryption keys may all be the nonreversible transformation of the same key. For example, the PIN encryption keys corresponding to encryption counter values:

...0110 0101 0100 0001

...0110 0101 0100 0010

...0110 0101 0100 0100

...0110 0101 0100 1000

...0110 0101 0101 0000

...0110 0101 0110 0000

are all nonreversible transformations of the PIN encryption key corresponding to encryption counter value:

...0110 0101 0100 0000.

Each of the six above-listed encryption counter values has a unique PIN encryption key associated with it, even though all six resulting PIN encryption keys are based on the same key. To describe how this is accomplished, we will define:

K-A as the key associated with …0110 0101 0100 0000.

and:

K-1 as the key associated with ...0110 0101 0100 0001

K-2 as the key associated with ...0110 0101 0100 0010

K-3 as the key associated with ...0110 0101 0100 0100

K-4 as the key associated with ...0110 0101 0100 1000

K-5 as the key associated with ...0110 0101 0101 0000

K-6 as the key associated with ...0110 0101 0110 0000

then:

K-1 = ...0110 0101 0100 0001 encrypted under K-A.

K-2 = ...0110 0101 0100 0010 encrypted under K-A.

K-3 = ...0110 0101 0100 0100 encrypted under K-A.

K-4 = ...0110 0101 0100 1000 encrypted under K-A.

K-5 = ...0110 0101 0101 0000 encrypted under K-A.

K-6 = ...0110 0101 0110 0000 encrypted under K-A.

In this way, K-1 through K-6 are each unique keys, but each is a nonreversible transformation of the same key, K-A.

Note that K-A will be used and thus is erased from the PIN Entry Device before any of the keys K-1 through K-6 are used. K-1 through K-6 (in this example) are generated and stored for future use before K-A is used and erased. Therefore, the PIN Entry Device is capable of storing a number of

such future keys. The number of future keys that the PIN Entry Device stores equals the number of binary bits in the encryption counter.

With this scheme, it is relatively easy for the Acquirer's security module to derive any PIN encryption key given its associated encryption counter value. After first deriving the PIN encryption key initially loaded in the PIN Entry Device, the module needs to perform only as many encryption operations as there are "one" bits in the encryption counter value. For example, assume that the security module receives an encrypted PIN block along with an encryption counter value of 1010 1100 0010. (A shorter-than-actual counter is used to simplify the example.) Assuming that the security module determines that the initial PIN encryption key was "K-I", the security module then proceeds with the following steps:

1) Encrypts 1000 0000 0000 using the PIN encryption key K-I.

2) Encrypts 1010 0000 0000 using the result of Step 1.

3) Encrypts 1010 1000 0000 using the result of Step 2.

4) Encrypts 1010 1100 0000 using the result of Step 3.

5) Encrypts 1010 1100 0010 using the result of Step 4.

Thus, in this example, the security module has determined the PIN encryption key used to encrypt the PIN in question in only five encryption cycles.

In order for the PIN Entry Device to be capable of up to 1 million encryptions, a 20-bit encryption counter could be used. This would require a maximum of 20 encryption cycles in the event that all 20 counter bits were "one". However, by going to a 21-bit encryption counter, the maximum number of encryption cycles can be reduced to 10. There are more than 2 million possible values for a 21-bit counter, but more than 1 million of these values have 10 or fewer "one" bits. Thus, in the specified algorithm, the encryption counter is incremented in such a way that no more than 10 "one" bits will ever occur simultaneously.

It should be noted that the cryptographic keys produced in a derived unique key per transaction system are 128-bit pseudo-random values, not 112-bit values with parity bits. In order to take cryptographic advantage of these 16 additional bits, a Non-reversible Key Generation Process has been specified. This function is believed to significantly increase the effort required to determine a key by an exhaustive "trial-and-error" procedure.

## A.4 DUKPT Test Data Examples

The following examples show how the encryption counter advances, the resulting PIN encryption keys with PIN blocks, and the MAC Encryption Keys and MAC values. Two sequences are provided: the first sequence shows the progression from initial load; the second shows the progression up to setting the encryption counter's MSB.

These sequences are based on the following data:

| | |
|---|---|
| Derivation Key: | 0123456789ABCDEFFEDCBA9876543210 |
| Initially Loaded Key Serial Number (KSN): | FFFF9876543210E00000 |
| Initially Loaded PIN Entry Device Key: | 6AC292FAA1315B4D 858AB3A3D7D5933A |
| Assumed PIN: | 1234 |
| Assumed Primary Account Number: | 4012345678909 |
| Formatted PIN | 041234FFFFFFFFFF |

| | |
|---|---|
| Clear ANSI PIN Block: | 041274EDCBA9876F |
| MAC and Data Encryption Input (ASCII) | "4012345678909D987" |
| MAC and Data Encryption Input (padded hex) | 3430313233343536.3738393039443938.3700000000000000 |

The examples in the following tables use hexadecimal representation for all keys and for the encrypted results. These examples are grouped into three rows for each Key Serial Number (KSN). The KSN and the two halves of the Generated Key are shown on the top row. The second row of the group shows the two halves of the PIN Encryption Key variant followed by the encrypted PIN Block (EPB) in the last column. The third row of each group shows the two halves of the MAC Encryption Key variant followed by the MAC result in the last column.

The MAC operations follow the CBC procedure described in ISO 16609 section C.4 using padding method 1 defined in ISO 9797 section 6.1.1. Here is an explanation of the steps:

1) Convert each hex digit (4012345678909D987) to the appropriate ASCII value:
34 30 31 32 33 34 35 36 37 38 39 30 39 44 39 38 37

2) XOR the Initial Vector (all zeros) with the first 8 bytes of the hex value from step 1:
3430313233343536

3) Encrypt the result from step 2 with the left half of the MAC Encryption Key.

4) XOR the result from step 3 with the next 8 bytes of the hex value from step 1. If fewer than 8 bytes remain, left justify the value and fill the remaining places with binary zeros.

5) Encrypt the result from step 4 with the left half of the MAC Encryption Key.

6) Go to step 4 until all input data has been used. After the last block has been encrypted, go to step 7.

7) Decrypt the result from step 5 with the right half of the MAC Encryption Key.

8) Encrypt the result from step 7 with the left half of the MAC Encryption Key.

9) Use the first 4 bytes of the result from step 8 as the MAC value. Show this result as hexadecimal digits.

### A.4.1 Variants of the Current Key

In section A.2 "Request PIN Entry 2", a description is provided for creating the PIN Encryption key variant. The PIN Encryption key is calculated as a variant of the current key as illustrated in Figure A-1 using the variant constant shown in Table A-1.

Section A.2 also includes a description of the calculation of MAC keys, applicable if the derived-unique-key-per-transaction PIN Entry Device performs the optional message authentication. One or two MAC keys are calculated as variants of the current key using the method illustrated in Figure A-1 and the applicable variant constants shown in Table A-1. MACing of the data follows algorithm 3 as described in ISO 16609 section C.4 using padding method 1 defined in ISO 9797 section 6.1.1.

Section A.2 also includes a description of the calculation of Data Encryption keys, applicable if the derived-unique-key-per-transaction PIN Entry Device performs the optional general-purpose data encryption. One or two data encryption keys are derived from the current key using the method illustrated in Figure A-2 and the applicable variant constants shown in Table A-1. In order to have

a high degree of isolation between the Data Encrypting key and the PIN Encrypting key, the Data Encrypting key is processed by a One Way Function (OWF) before use. The OWF defined has the derived variant value encrypted using itself as the key. The values "Variant base-L" and "Variant base-R" are not parity adjusted before they are used as input to the One Way Function. Encryption of the data should use T-DEA in CBC mode.

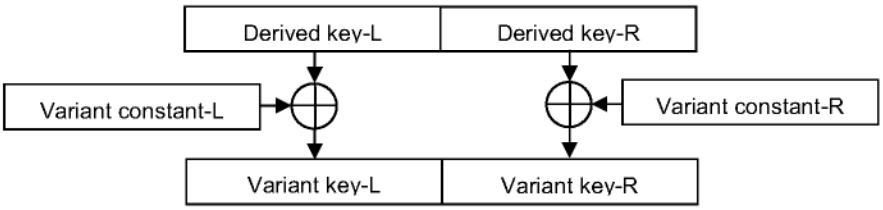



**Figure A-1 – Key calculation for PIN-encrypting key and MAC keys**

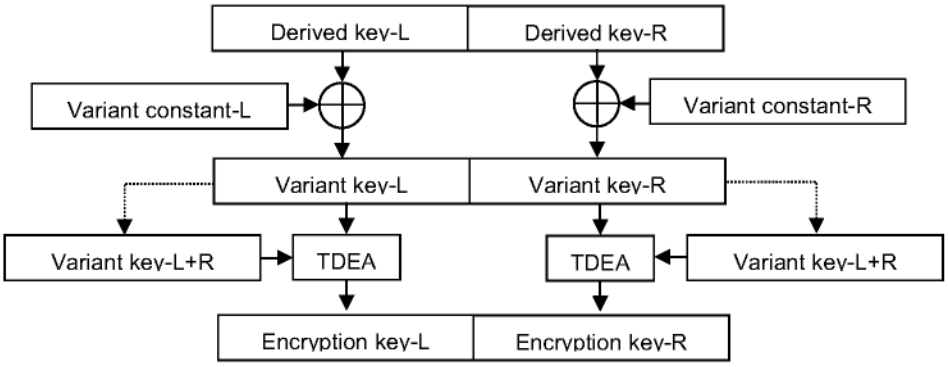



**Figure A-2 – Key calculation for Data Encryption keys**

| Key used for | Variant constant | |
|---|---|---|
| | Variant constant-L | Variant constant-R |
| PIN Encryption | 00 00 00 00 00 00 00 FF | 00 00 00 00 00 00 00 FF |
| Message Authentication, request or both ways | 00 00 00 00 00 00 FF 00 | 00 00 00 00 00 00 FF 00 |
| Message Authentication, response | 00 00 00 00 FF 00 00 00 | 00 00 00 00 FF 00 00 00 |
| Data Encryption, request or both ways | 00 00 00 00 00 FF 00 00 | 00 00 00 00 00 FF 00 00 |
| Data Encryption, response | 00 00 00 FF 00 00 00 00 | 00 00 00 FF 00 00 00 00 |

**Table A-1 - Variant constants for transaction keys**

Note: This table is a summary of the variants that may be used to modify the current key register to create separate keys for separate functions. The Data Encryption variants are the only variants that includes the use of a One Way Function (see Figure A-2 above) to generate that the separate keys which may be used in the request and the response.

### A.4.2 Initial Sequence

| KSN | Generated Key L<br>PIN Encryption Key L<br>MAC (Request) Key L<br>MAC (Response) Key L<br>Data Encryption Key L | Generated Key R<br>PIN Encryption Key R<br>MAC (Request) Key R<br><br>MAC (Response) Key R<br>Data Encryption Key R | EPB<br>MAC (Request)<br><br>MAC (Response)<br>Encrypted Data |
|---|---|---|---|
| FFFF9876543210E00001 | 042666B49184CFA3 | 68DE9628D0397BC9 | |
| PIN | 042666B49184CF5C | 68DE9628D0397B36 | 1B9C1845EB993A7A |
| MAC (Request) | 042666B4918430A3 | 68DE9628D03984C9 | 9CCC78173FC4FB64 |
| MAC (Response) | 042666B46E84CFA3 | 68DE96282F397BC9 | 20364223C1FF00FA |
| DATA | 448D3F076D830403 | 6A55A3D7E0055A78 | FC0D53B7EA1FDA9E.E68AAF2E70D9B950.6229BE2AA993F04F |
| FFFF9876543210E00002 | C46551CEF9FD24B0 | AA9AD834130D3BC7 | |
| PIN | C46551CEF9FD244F | AA9AD834130D3B38 | 10A01C8D02C69107 |
| MAC (Request) | C46551CEF9FDDBB0 | AA9AD834130DC4C7 | F608A9BCA6FFC311 |
| MAC (Response) | C46551CE06FD24B0 | AA9AD834EC0D3BC7 | D1FCA6BEF05D24D2 |
| DATA | F1BE73B36135C5C2 | 6CF937D50ABBE5AF | A2B4E70F846E63D6.8775B7215EB4563D.FD3037244C61CC13 |
| FFFF9876543210E00003 | 0DF3D9422ACA56E5 | 47676D07AD6BADFA | |
| PIN | 0DF3D9422ACA561A | 47676D07AD6BAD05 | 18DC07B94797B466 |
| MAC (Request) | 0DF3D9422ACAA9E5 | 47676D07AD6B52FA | 20B59A4FEAC937E3 |
| MAC (Response) | 0DF3D942D5CA56E5 | 47676D07526BADFA | BAD4CC9CC2AE326C |
| DATA | EEEEF522C67239E4 | A2A65FEBF4C511F4 | BD751E65F10E75B6.C1D5B1D283496A36.C2DE21D993C387A7 |
| FFFF9876543210E00004 | 279C0F6AEED0BE65 | 2B2C733E1383AE91 | |
| PIN | 279C0F6AEED0BE9A | 2B2C733E1383AE6E | 0BC79509D5645DF7 |
| MAC (Request) | 279C0F6AEED04165 | 2B2C733E13835191 | C7BFA6CC44161828 |
| MAC (Response) | 279C0F6A11D0BE65 | 2B2C733EEC83AE91 | 1EB08AEECE6FF0C2 |
| DATA | BCF2610997C3AC3C | 5F13AE965A1B773B | 1118F50947441BBD.A3C8C70220021A12.EC31CC473F7215F4 |
| FFFF9876543210E00005 | 5F8DC6D2C845C125 | 508DDC048093B83F | |
| PIN | 5F8DC6D2C845C1DA | 508DDC048093B8C0 | 5BC0AF22AD87B327 |
| MAC (Request) | 5F8DC6D2C8453E25 | 508DDC048093473F | 0202B96339022058 |
| MAC (Response) | 5F8DC6D23745C125 | 508DDC047F93B83F | 5CBE3E81D1D2A0FB |
| DATA | F3054D8B7471284B | DB4EE18AFC3B091B | 9FD7BD1EC28845AC.A93367A9DA9317BD.555C6B33AE22D365 |

| | | | |
|---|---|---|---|
| FFFF9876543210E00006 | 5E415CB0BAF9F03C | D0C14B63FB62FF43 | |
| PIN | 5E415CB0BAF9F0C3 | D0C14B63FB62FFBC | A16DF70AE36158D8 |
| MAC (Request) | 5E415CB0BAF90F3C | D0C14B63FB620043 | CF6C72E6A49892D5 |
| MAC (Response) | 5E415CB045F9F03C | D0C14B630462FF43 | C60D38E858C936EB |
| DATA | BAF444BBC5D02752 | D6CEC9C51226AF53 | DC526613AD9095C1.CEC60C41B7686ED6.06AC1AA2F4F54912 |
| FFFF9876543210E00007 | 0C8F780B7C8B49D0 | AE84A9EB2A6CE660 | |
| PIN | 0C8F780B7C8B492F | AE84A9EB2A6CE69F | 27711C16CB257F8E |
| MAC (Request) | 0C8F780B7C8BB6D0 | AE84A9EB2A6C1960 | B11EB0D97CF167E8 |
| MAC (Response) | 0C8F780B838B49D0 | AE84A9EBD56CE660 | 053EFD396A0B1333 |
| DATA | E613858B288B6ACA | A7AE454931B21C56 | 24700BD6F1775153.1F2A16CE2AF77311.01E6F87839C67244 |
| FFFF9876543210E00008 | 27F66D5244FF62E1 | AA6F6120EDEB4280 | |
| PIN | 27F66D5244FF621E | AA6F6120EDEB427F | 50E55547A5027551 |
| MAC (Request) | 27F66D5244FF9DE1 | AA6F6120EDEBBD80 | 3679055BCCBE3D6B |
| MAC (Response) | 27F66D52BBFF62E1 | AA6F612012EB4280 | 7FACA91D970D9187 |
| DATA | C39B2778B058AC37 | 6FB18DC906F75CBA | 5F6584DEAFC51B48.A239B51C2DACA971.76C01D9CA8EDC33C |
| FFFF9876543210E00009 | 27E31064FDC56569 | 8900E2057F658E7E | |
| PIN | 27E31064FDC56596 | 8900E2057F658E81 | 536CF7F678ACFC8D |
| MAC (Request) | 27E31064FDC59A69 | 8900E2057F65717E | 26AA23DC169152F8 |
| MAC (Response) | 27E3106402C56569 | 8900E20580658E7E | 37F17B8F302EE3A6 |
| DATA | BC6612B3A4FAB5B1 | 65AC928BBD9DB6F0 | 7CB9080923ED4D7D.8D1B8A2849331684.CC910522023BE537 |
| FFFF9876543210E0000A | 6CF2500A22507C7C | C776CEADC1E33014 | |
| PIN | 6CF2500A22507C83 | C776CEADC1E330EB | EDABBA23221833FE |
| MAC (Request) | 6CF2500A2250837C | C776CEADC1E3CF14 | 1632621C039098CE |
| MAC (Response) | 6CF2500ADD507C7C | C776CEAD3EE33014 | CE038CC91CB4CBBE |
| DATA | BC9C848366B1F999 | 7656DD8004C5C43B | C221F5A8CA5DA909.BE04938E0FB8DFCF.B30ED0D62B590663 |
| FFFF9876543210E0000B | 3E8260BA04B2D620 | C01482B3819A18B7 | |
| PIN | 3E8260BA04B2D6DF | C01482B3819A1848 | 2328981C57B4BDBA |
| MAC (Request) | 3E8260BA04B22920 | C01482B3819AE7B7 | 76D95CB4F6B0D093 |
| MAC (Response) | 3E8260BAFBB2D620 | C01482B37E9A18B7 | 27C9FA5E14F0A044 |
| DATA | 0D790AC0C7E16699 | F235293F546D29CD | 98A271C13E4AEE80.C6343867F2F0F6C5.5E8B7E5645866A0B |
| FFFF9876543210E0000C | B716E1E11CF53D80 | 726CAEE75C3A624F | |
| PIN | B716E1E11CF53D7F | 726CAEE75C3A62B0 | 038D03CC926CF286 |
| MAC (Request) | B716E1E11CF5C280 | 726CAEE75C3A9D4F | BAD9D84D1C934D3B |

| | | | |
|---|---|---|---|
| MAC (Response) | B716E1E1E3F53D80 | 726CAEE7A33A624F | 6D2ECC6A3CE2A378 |
| DATA | D57E353ABF3F0CB0 | B951B5D8820AD741 | 273CBE2F275B1C29.D53F3C01A24DEC73.F92C3BE219FFCB91 |
| FFFF9876543210E0000D | E072EDF95340535F | B6C581C58FBF2533 | |
| PIN | E072EDF9534053A0 | B6C581C58FBF25CC | 6C8AA97088B62C68 |
| MAC (Request) | E072EDF95340AC5F | B6C581C58FBFDA33 | AAAFB5983360C89A |
| MAC (Response) | E072EDF9AC40535F | B6C581C570BF2533 | 7E40B49274596E49 |
| DATA | ACAEAA2E23F99005 | 52C2132890CE5390 | 7290CB9D6F6E35A9.B8A34C590A93696F.8FE0137A8558884C |
| FFFF9876543210E0000E | A80046087F5B8F24 | 5AAD95E18429084F | |
| PIN | A80046087F5B8FDB | 5AAD95E1842908B0 | F17C9E1D72CD4950 |
| MAC (Request) | A80046087F5B7024 | 5AAD95E18429F74F | 4EC79532D8EC0E30 |
| MAC (Response) | A8004608805B8F24 | 5AAD95E17B29084F | 552339891A346CA4 |
| DATA | F42D46659549ED0A | DADF6AB332815A7B | 48207931E5F6E5E3.E02E7E1D7BED4E32.32FCA67E66A31F61 |
| FFFF9876543210E0000F | 93DD5B956C487847 | 2E453AAEFD32A5AA | |
| PIN | 93DD5B956C4878B8 | 2E453AAEFD32A555 | B170F6E7F7F2F64A |
| MAC (Request) | 93DD5B956C488747 | 2E453AAEFD325AAA | 82500D05251841E7 |
| MAC (Response) | 93DD5B9593487847 | 2E453AAE0232A5AA | AFE23CBA2307F7CC |
| DATA | F331F347ADD182B7 | CDC5208F62CA1233 | 166246602B125511.4A4012513B0DD4A9.E19E82C945860C55 |
| FFFF9876543210E00010 | 59598DCBD9BD94C0 | 94165CE453585F57 | |
| PIN | 59598DCBD9BD943F | 94165CE453585FA8 | D5D9638559EF53D6 |
| MAC (Request) | 59598DCBD9BD6BC0 | 94165CE45358A057 | D39D3A598549B9CD |
| MAC (Response) | 59598DCB26BD94C0 | 94165CE4AC585F57 | 9E933C88DA838A4F |
| DATA | 82D613B1742B19C5 | 5F653616A47DF1A0 | 53EDA44BDBCA037B.2D3FEAEBAF734993.36767BD1D0D09679 |
| FFFF9876543210E00011 | 2B5F01F4F0CC05EA | 639D523231BF1BA2 | |
| PIN | 2B5F01F4F0CC0515 | 639D523231BF1B5D | D544F8CDD292C863 |
| MAC (Request) | 2B5F01F4F0CCFAEA | 639D523231BFE4A2 | F094AF9309CD8A91 |
| MAC (Response) | 2B5F01F40FCC05EA | 639D5232CEBF1BA2 | 631E7B39C609220B |
| DATA | 5090137BD2EE8305 | 88765B84F7DA9F91 | 086F3F46238B99AF.B35DA42029675EE1.FAB1C606307B35FB |
| FFFF9876543210E00012 | 9CF640F279C2AEE6 | 15F725EEEAC2CBAF | |
| PIN | 9CF640F279C2AE19 | 15F725EEEAC2CB50 | 7A21BD10F36DC41D |
| MAC (Request) | 9CF640F279C251E6 | 15F725EEEAC234AF | A819AF04330EB313 |
| MAC (Response) | 9CF640F286C2AEE6 | 15F725EE15C2CBAF | E93F5E271A20D156 |
| DATA | 91E5A8B18169A953 | 2DF547D6AFE6087C | 270439430E60BFC6.00FE3DC7E7AD8C72.D22D5284EC676EF2 |
| FFFF9876543210E00013 | C3DF489FDF1153B4 | F03DE97C27DC4C2F | |

| | | | |
|---|---|---|---|
| PIN | C3DF489FDF11534B | F03DE97C27DC4CD0 | 78649BD17D0DFA60 |
| MAC (Request) | C3DF489FDF11ACB4 | F03DE97C27DCB32F | 79285A1733915039 |
| MAC (Response) | C3DF489F201153B4 | F03DE97CD8DC4C2F | FCE7A5A5A7333665 |
| DATA | 44893E3434ABDD6A | 817CE2841825E1FD | 2FEAE132FA0E46B4.ABF55A87477AFA90.7E598868BCC236DA |
| FFFF9876543210E00014 | 658488507721B3F1 | 4737FA93F923CBD2 | |
| PIN | 658488507721B30E | 4737FA93F923CB2D | 7E7E16EA0C31AD56 |
| MAC (Request) | 6584885077214CF1 | 4737FA93F92334D2 | E2B2637FC4BB5D0D |
| MAC (Response) | 658488508821B3F1 | 4737FA930623CBD2 | A10E94AA46F9FA03 |
| DATA | E0D4C7B3EA32A8F1 | 08A83C1844ED6FD8 | A16A61328E36A218.58B5FBFBE7BAF798.5F5C54F5B8C12298 |
| FFFF9876543210E00015 | E161D1956A61F6D2 | F37AFD7F9CC3699A | |
| PIN | E161D1956A61F62D | F37AFD7F9CC36965 | 72105C22EBC791E6 |
| MAC (Request) | E161D1956A6109D2 | F37AFD7F9CC3969A | B2033EBD58DA8BF1 |
| MAC (Response) | E161D1959561F6D2 | F37AFD7F63C3699A | BDA3AEE182D0FFEA |
| DATA | D9AE3E62F5E3CA2C | 357E37F500D9F314 | 42A9342FD5C1ACD2.0E137950A7368D2B.BE884275B7906918 |

### A.4.3 MSB Rollover Sequence

| | | | |
|---|---|---|---|
| FFFF9876543210EFF800 | F9CDFEBF4F5B1D9E | B3EC12454527E176 | |
| PIN | F9CDFEBF4F5B1D61 | B3EC12454527E189 | 33365F5CC6F23C35 |
| MAC (Request) | F9CDFEBF4F5BE29E | B3EC124545271E76 | 23E3CB62478E6EA4 |
| MAC (Response) | F9CDFEBFB05B1D9E | B3EC1245BA27E176 | 5776C37671CD7AF9 |
| DATA | A9E03F541E29B4FD | D47141AECF25190B | 5FF8C648565975E7.46C0CB7B1157DCB6.EC3C5F25E82F8A62 |
| FFFF9876543210EFF801 | 5BEE92627E97825C | 911BF619DF72CA3B | |
| PIN | 5BEE92627E9782A3 | 911BF619DF72CAC4 | 3A86BF003F835C9D |
| MAC (Request) | 5BEE92627E977D5C | 911BF619DF72353B | 97623632CAD07B98 |
| MAC (Response) | 5BEE92628197825C | 911BF6192072CA3B | 69D1EF4762ACC916 |
| DATA | EBC9A12BDFAFC470 | E00ECA54611BE5C9 | 55374FDBB3B2B4ED.0751243FC0C54DB1.F3F3CF76F0C0B3C4 |
| FFFF9876543210EFF802 | 64CA2F8AB93F61B0 | 1620B00BF77C7301 | |
| PIN | 64CA2F8AB93F614F | 1620B00BF77C73FE | 3DB977D05C36DF3F |
| MAC (Request) | 64CA2F8AB93F9EB0 | 1620B00BF77C8C01 | 726085213E40251C |
| MAC (Response) | 64CA2F8A463F61B0 | 1620B00B087C7301 | BF7036B7CE0467D4 |
| DATA | C27EE6571C9855AE | 5DEA206A0C6FCCE8 | 74A1250E3234DEDE.B422FA7F862C484B.BA6924DF9BC32CE5 |
| FFFF9876543210EFF804 | 35A168695B3BA833 | 07E871C42021DD37 | |
| PIN | 35A168695B3BA8CC | 07E871C42021DDC8 | BA83243305712099 |
| MAC (Request) | 35A168695B3B5733 | 07E871C420212237 | D86687B77B29ADC9 |
| MAC (Response) | 35A16869A43BA833 | 07E871C4DF21DD37 | 645A0A18A6F318AF |
| DATA | 069B700F504E2C53 | 313D860CD34AEC9A | 159E9FA8E979A9C7.6C118C7BE9E5D436.AC0B22CD098BAE1F |
| FFFF9876543210EFF808 | 12C380FCB72ED3C2 | 6FC6079AF63F4787 | |
| PIN | 12C380FCB72ED33D | 6FC6079AF63F4778 | B0DA04AC90A36D85 |
| MAC (Request) | 12C380FCB72E2CC2 | 6FC6079AF63FB887 | 5E00713D3827FC2F |
| MAC (Response) | 12C380FC482ED3C2 | 6FC6079A093F4787 | E8D8026A95972BFB |
| DATA | 512FCA997E957FAE | 5E47C6EC4F635BAA | F87F61877DFF1CD3.9A635BEA4D15F81D.30793B9157805BE4 |
| FFFF9876543210EFF810 | D8007C833A1E3C49 | C6889B97A68C54FF | |
| PIN | D8007C833A1E3CB6 | C6889B97A68C5400 | 2CF02BD9C309EEDA |
| MAC (Request) | D8007C833A1EC349 | C6889B97A68CABFF | 1BCCF968BA8707E9 |
| MAC (Response) | D8007C83C51E3C49 | C6889B97598C54FF | 9623FF4581EC1166 |
| DATA | D4E324DBF3FA15CF | 72A3EC6D7F8D9A6B | 4DEA2DDA8D079F80.40CAD885C29A4037.E9490480256D074D |
| FFFF9876543210EFF820 | A9C9FF9B735FDD49 | 591F296B0BF0FD70 | |

| | | | |
|---|---|---|---|
| PIN | A9C9FF9B735FDDB6 | 591F296B0BF0FD8F | 9D1E2F77AEEE81C6 |
| MAC (Request) | A9C9FF9B735F2249 | 591F296B0BF00270 | 2F6D2191FD75B647 |
| MAC (Response) | A9C9FF9B8C5FDD49 | 591F296BF4F0FD70 | FF864D190E4CE105 |
| DATA | C7563DC6EAF85B1E | BE23D6E96AD02E05 | CE625FFBBB339C83.05FA86CBFD192F42.92B29962649947DD |
| FFFF9876543210EFF840 | 0E3E221062CB5E03 | A384602783A004FA | |
| PIN | 0E3E221062CB5EFC | A384602783A00405 | 40870B0F8BA2011C |
| MAC (Request) | 0E3E221062CBA103 | A384602783A0FBFA | 4DA4CBAC62677694 |
| MAC (Response) | 0E3E22109DCB5E03 | A38460277CA004FA | 3CE7AD5052AC3AE9 |
| DATA | C0D97706DEEC392D | D935C20114D2616A | 29DDF3F2C3B9BE76.A8338E2B22BA668B.68B46AAE9C714607 |
| FFFF9876543210EFF880 | CA8225C587124DFB | B87F279C0BE29727 | |
| PIN | CA8225C587124D04 | B87F279C0BE297D8 | 22E340D6ABB40981 |
| MAC (Request) | CA8225C58712B2FB | B87F279C0BE26827 | 95C43BC2C019DBAC |
| MAC (Response) | CA8225C578124DFB | B87F279CF4E29727 | 73D03F0B90E6DBA1 |
| DATA | 9CDC6AB64EF49603 | 1C5B42C0072F55AC | 9E9AA50594849DEB.63BC3EF3BE63C563.7F03D03B0DCFA973 |
| FFFF9876543210EFF900 | 906AAC1751D4971A | 144D52C76C6E1F97 | |
| PIN | 906AAC1751D497E5 | 144D52C76C6E1F68 | 1A4C10AFBA03A430 |
| MAC (Request) | 906AAC1751D4681A | 144D52C76C6EE097 | D6BEAA71DF19FF0A |
| MAC (Response) | 906AAC17AED4971A | 144D52C7936E1F97 | 5D461B7199531F3D |
| DATA | 516A8A6BB6B32497 | CC755050F9E0FD3A | C653AFCD3245ADB3.B113FFC48B46BD0B.6C5B3B5FB10044EE |
| FFFF9876543210EFFA00 | 2B9E2CE7C2750337 | E10B54D2989EA6AF | |
| PIN | 2B9E2CE7C27503C8 | E10B54D2989EA650 | 849763B43E5F9CFF |
| MAC (Request) | 2B9E2CE7C275FC37 | E10B54D2989E59AF | 93930C9F4AEC91D7 |
| MAC (Response) | 2B9E2CE73D750337 | E10B54D2679EA6AF | 531DA5F91D4E6C7C |
| DATA | D7B1D0921FE592A4 | F7BBF84A55D044CB | 50EB117D8CB2BE5A.DBBE98B6570C557C.13622688A61D559D |
| FFFF9876543210EFFC00 | F9430DF975082491 | C77BE4EF4FDB91EE | |
| PIN | F9430DF97508246E | C77BE4EF4FDB9111 | DEFC6F09F8927B71 |
| MAC (Request) | F9430DF97508DB91 | C77BE4EF4FDB6EEE | 2917C6EFF391D1BF |
| MAC (Response) | F9430DF98A082491 | C77BE4EFB0DB91EE | AA8D2DA65925AE3B |
| DATA | D6E493D5A769EC96 | 1FF420E770C9A4A3 | A2C773BC45B277E3.F100C37E30A6A6FB.1865134815B05EBC |
| FFFF9876543210F00000 | AA4D58DB653EC74A | 48C75F2F047DD2B5 | |
| PIN | AA4D58DB653EC7B5 | 48C75F2F047DD24A | 73EC88AD0AC5830E |
| MAC (Request) | AA4D58DB653E384A | 48C75F2F047D2DB5 | 7DFA93155E2D87A0 |
| MAC (Response) | AA4D58DB9A3EC74A | 48C75F2FFB7DD2B5 | 1444BEC3726A3EA6 |

| | | | |
|---|---|---|---|
| DATA | FE4BFE999550B26C | C7D6A3602046612D | 1E830E7F53FC171E.35380145320233EF.9C6F3B08CED984AB |

### A.4.4 Calculation and Storage of DUKPT Transaction Keys at the Terminal

Table A-2 below illustrates the changing key values stored in the 21 Future Key Registers (FKR) as the Transaction Counter (TC) increases.

Where an FKR contains a numeric value (m) at a particular value of the TC, it indicates that a new key value is derived and stored in the FKR at that transaction. The value m is used as the data in the key derivation, and is equal to the value of the transaction counter when the key value will be extracted and used as the current transaction key. The value m is a function of the current transaction counter (n) and the FKR number (r), where:

$$m = n + 2^{21-r}.$$

Initially (at time n = 0) all 21 FKRs are initialized using the initial key (IK) as the derivation key.

Subsequently, where an FKR contains the character **X** at a particular value of the TC, it indicates:

extract the key from this FKR and use variants of it for PIN encryption, MACing etc. in this transaction.

if TC incorporates less than ten 1 bits-

use the extracted key to derive future keys and store them in the higher-numbered FKRs.

increment TC by 1.

if TC incorporates ten 1 bits-

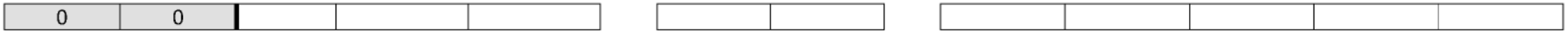
increment TC by $2^{21-r}$ where r is the number of the FKR (i.e. add 1 to the least significant 1 bit of TC).

delete the extracted key.

The process above causes values of TC that incorporate more than ten 1 bits to be skipped. The table incorporates the TC values in hex in order to help see those values. The more heavily shaded values cause a skip in TC. (Note: Lower values than 7FE have ten 1 bits set, but they are all odd numbers and so do not cause a skip in TC, e.g. 3FF, 5FF, 6FF … 7FD)

**Table A-2 Chronological Accesses to Future Key Registers**

| FKR | | 1 | 2 | 3 | | 10 | 11 | | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TC | | | | | | | | | | | | | |
| (dec) | (hex) | | | | | | | | | | | | |
| 0 (init) | 0 | $2^{20}$ | $2^{19}$ | $2^{18}$ | ... | $2^{11}$ | $2^{10}$ | ... | 16 | 8 | 4 | 2 | 1 |
| 1 | 1 | | | | | | | | | | | | X |
| 2 | 2 | | | | | | | | | | | X | 3 |
| 3 | 3 | | | | | | | | | | | | X |
| 4 | 4 | | | | | | | | | | X | 6 | 5 |
| 5 | 5 | | | | | | | | | | | | X |
| 6 | 6 | | | | | | | | | | | X | 7 |
| 7 | 7 | | | | | | | | | | | | X |
| 8 | 8 | | | | | | | | | **X** | 12 | 10 | 9 |
| 9 | 9 | | | | | | | | | | | | **X** |
| 10 | A | | | | | | | | | | | **X** | 11 |
| 11 | B | | | | | | | | | | | | **X** |
| 12 | C | | | | | | | | | | **X** | 4 | 13 |
| 13 | D | | | | | | | | | | | | **X** |
| 14 | E | | | | | | | | | | | **X** | 15 |
| 15 | F | | | | | | | | | | | | **X** |
| 16 | 10 | | | | ... | | | ... | **X** | 24 | 20 | 18 | 17 |
| 17 | 11 | | | | | | | | | | | | **X** |
| | | | | | | | | | | | | | |
| 2045 | 7FD | | | | | | | | | | | | **X** |
| 2046 | 7FE | | | | | | | | | | | **X** | |
| 2048 | 800 | | | | | **X** | 3072 | | 2064 | 2056 | 2052 | 2050 | 2049 |
| 2049 | 801 | | | | | | | | | | | | **X** |
| | | | | | | | | | | | | | |
| 1047552 | FFC00 | | | | | | **X** | | | | | | |
| 1048576 (=$2^{20}$) | 100000 | **X** | $2^{20}+2^{19}$ | $2^{20}+2^{18}$ | | $2^{20}+2^{11}$ | $2^{20}+2^{10}$ | | $2^{20}+16$ | $2^{20}+8$ | $2^{20}+4$ | $2^{20}+2$ | $2^{20}+1$ |
| $2^{20}+1$ | 100001 | | | | | | | | | | | | **X** |
| | | | | | | | | | | | | | |
| 2095104 | 1FF800 | | | | | **X** | | | | | | | |



## A.5 "Security Module" Algorithm For Automatic PIN Entry Device Checking

To ensure that a PIN Entry Device has been properly designed, it is desirable to exhaustively test a unit through all 1 million possible keys.

The temporary storage areas are defined as follows:

**R8:** **8-byte register.**

**R8A:** **8-byte register.**

**R8B:** **8-byte register**

**R3:** **21-bit register.**

**SR:** **21-bit shift register.**

**KSNR:** **8-byte register; right-most 8 bytes of the Key Serial Number as received from the PIN Entry Device.**

**IKEY:** **16-byte register; PIN encryption key initially loaded into the PIN Entry Device.**

**CURKEY:** **16-byte register; at completion of the algorithm, it contains the PIN encryption key used in the encryption of the PIN in the current transaction.**

The processing algorithm is as follows:

1) Copy IKEY into CURKEY.

2) Copy KSNR into R8.

3) Clear the 21 right-most bits of R8.

4) Copy the 21 right-most bits of KSNR into R3.

5) Set the left-most bit of SR, clearing the other 20 bits.

**"TAG1"**

1) Is SR AND'ed with R3 = 0? If yes, go to "TAG2".

2) "OR" SR into the 21 right-most bits of R8. (This sets the R8 bit corresponding to the SR bit that is set.)

3) XOR the right half of CURKEY with R8 and store the result into R8A.

4) DEA-encrypt R8A using the left half of CURKEY as the key and store the result into R8A.

5) XOR R8A with the right half of CURKEY and store the result into R8A.

6) XOR CURKEY with hexadecimal C0C0 C0C0 0000 0000 C0C0 C0C0 0000 0000.

7) XOR the right half of CURKEY with R8 and store the result into R8B.

8) DEA-encrypt R8B using the left half of CURKEY as the key and store the result into R8B.

9) XOR R8B with the right half of CURKEY and store the result into R8B.

10) Store R8A into the right half of CURKEY.

11) Store R8B into the left half of CURKEY.

**"TAG2"**

1) Shift SR right one bit.

2) If SR is not equal to zero (if the "one" bit has not been shifted off), go to "TAG1".

3) XOR CURKEY with hexadecimal "0000 0000 0000 00FF 0000 0000 0000 00FF" and go to "Exit". (CURKEY now holds the PIN-encryption key that the security module will use to triple-DEA decrypt the received encrypted PIN block.)

## A.6 Derivation Of The Initial Key

Note: References to the base derivation key in this document deal only with double-length keys.

The initial PIN Entry Device key (the key initially loaded into the PIN Entry Device) is generated by the following process:

1) Copy the entire key serial number, including the 21-bit encryption counter, right-justified into a 10-byte register. If the key serial number is less than 10 bytes, pad to the left with hex "FF" bytes.

2) Set the 21 least-significant bits of this 10-byte register to zero.

3) Take the 8 most-significant bytes of this 10-byte register, and encrypt/decrypt/encrypt these 8 bytes using the double-length derivation key, per the TECB mode of Reference 2.

4) Use the ciphertext produced by Step 3 as the left half of the Initial Key.

5) Take the 8 most-significant bytes from the 10-byte register of Step 2 and encrypt/decrypt/encrypt these 8 bytes using as the key the double-length derivation key XORed with hexadecimal C0C0 C0C0 0000 0000 C0C0 C0C0 0000 0000, per the TECB mode of Reference 2.

6) Use the ciphertext produced by Step 5 as the right half of the Initial Key.

# Annex B
# (Informative)
# SMID Examples

(This Annex is not part of this standard and is included for information only.)

The SMID content is preceded with the LLL (not shown in the following examples) in accordance with Reference 7. In the examples, X'cc...' indicates characters representing hex digits. All other characters are ASCII representation unless otherwise indicated in the examples. In all applicable cases, field lengths are specified as the number of bytes.

1) Examples When Control Field Has Hex Value A1

The SMID is identifying a transaction key to be used with the Fixed Transaction Key method. In these examples, assume that the key name is specified as ASCII characters. Note there are no embedded length fields in the SMID.

- X'A1' JANUARY1990KEY23
- X'A1' MACKEY3
- X'A1' ACME
- X'A1' 1
- X'A1' 2

2) Examples When Control Field Has Hex Value A2

The SMID is identifying a transaction key to be used with the Master Key/ Transaction Key method. In these examples, assume that the key name is specified as ASCII characters. Note there are no embedded length fields in the SMID.

- X'A2' JANUARY1990KEY31
- X'A2' SAFE
- X'A2' EKD31
- X'A2' 3

3) Examples When Control Field Has Hex Value A5

The SMID is identifying a transaction key to be used with the DUKPT method and a control byte is used. In these examples, assume that the key name is specified as hex values. Note there are no embedded length fields in the SMID. The boundaries between the key set identifier, TRSM identifier, and Transaction Counter are either self-defining or implicitly defined.

- X'A5'X'77665544332211O123456FFEEDE00001'
- X'A5'X'234567987654000011'

- X'A5'X'234567987654000012'
- X'A5'X'234567987654000013'
- X'A5'X'41111111233333'
- X'A5'X'41111111233334'
- X'A5'X'5244447A3B'
- X'A5'X'5244447A3C'
- X'A5'X'5432104B3F27IC69AC2B'

4) Examples When Control Field Has Hex Value B1

The SMID is identifying a transaction key to be used with the Fixed Transaction Key method. In these examples, assume that the key name is specified as ASCII characters. Unlike in A1, note there are embedded length fields in the SMID.

- X'BI' X'10' JANUARY1990KEY23
- X'BI' X'07' MACKEY3
- X'BI' X'04' ACME
- X'BI' X'0I' 1
- X'BI' X'0I' 2

5) Examples When Control Field Has Hex Value B2

The SMID is identifying a transaction key to be used with the Master Key/ Transaction Key method. In these examples, assume that the key name is specified as ASCII characters. Unlike in A2, note there are embedded length fields in the SMID.

- X'B2' X'10' JANUARY1990KEY31
- X'B2' X'04' SAFE
- X'B2' X'05' EKD31
- X'B2' X'0I' 3

6) Examples When Control Field Has Hex Value B5

The SMID is identifying a transaction key to be used with the DUKPT method and a control byte is used. In these examples, assume that the key name is specified as hex values. Unlike in A5, note there are embedded length fields in the SMID. The boundaries between the key set identifier, TRSM identifier, and Transaction Counter are either self-defining or implicitly defined.

- X'B5' X'10' X'776655443322110123456FFEEDE00001'
- X'B5' X'09' X'234567987654000011'

- X'B5' X'09' X'234567987654000012'
- X'B5' X'09' X'234567987654000013'
- X'B5' X'07' X'41111111233333'
- X'B5' X'07' X'41111111233334'
- X'B5' X'05' X'5244447A3B'
- X'B5' X'05' X'5244447A3C'
- X'B5' X'0A' X'5432104B3F27lC69AC2B'

7) Examples When Control Field Has Hex Value C1

No examples, since C1 not valid.

8) Examples When Control Field Has Hex Value C2

The SMID is conveying, using the Master Key/Transaction Key method, security or key management information to be used in future transaction messages. In these examples, assume that the Cryptographic Service Messages (CSM) specified as ASCII characters. Note there are no embedded length fields in the SMID. The CSMs are as specified in Annex D.

- X'C2' CSM(MCL/KSM RCV/B ORG/A KD/0123456789ABCDEF.P.SAFE CTP/001 MAC/8765 4321)
  (node A distributes key SAFE, encrypted, to node B)
- X'C2' CSM(MCL/RSM RCV/A ORG/B MAC/FEDC BA98)
  (node B acknowledges receipt of last key, to node A)

9) Examples When Control Field Has Hex Value C3

No examples, since C3 not valid.

10) Examples When Control Field Has Hex Value C4

No examples, since C4 not valid.

11) Examples When Control Field Is Not Present

When the Control Field is omitted, the SMID is identifying the current transaction key. The method for determining the key is identified from a database entry indicated by the KEY SET IDENTIFIER, the left-most portion of the key name. It is recommended that the KEY SET IDENTIFIER be assigned as described in Annex E. The length of the KEY SET IDENTIFIER is self defining, and the host can identify the method for determining the key from a database entry indicated by the KEY SET IDENTIFIER.

- X'5432104B3F271C69AC2B' (Host recognizes "543210" as a 6-digit KEY SET IDENTIFIER. (Cannot have a 5-digit KEY SET IDENTIFIER of "54321", nor a 7-digit KEY SET IDENTIFIER of "543210X".) The database entry for "543210" specifies a specific DUKPT implementation and the Base Derivation Keys to be used. The "4B3F..." identifies the key within the key set.)
- X'4556677AC72C46199D' (Host recognizes "4556677" as a 7-digit KEY SET IDENTIFIER. The database entry for "4556677" specifies a specific Fixed Key implementation. The "AC72..." identifies the key within the key set.)

If the entire SMID is omitted, then something else in the transaction or node indicates how to determine the key or database entry. Some examples may be:

a) The terminal identifier may indicate that the Master Key/Transaction Key should be used.

b) The card acceptor knows that all its terminals employ only the Master Key/ Transaction Key.

c) The host knows that it always communicates with one fixed key.

d) The host communicates with one of ten fixed keys. A transaction containing a SMID identifies which key to use until further notice.

e) The host communicates with a transaction key established using Master Key/Transaction Key. A transaction containing a SMID identifies which key to use until further notice. A transaction (perhaps an administrative network message) containing a SMID establishes a new key value using a CSM.

12) Examples When Control Field Has Hex Value B0

The SMID is employing the special feature permitting several SMIDs to be conveyed in a single message. Note there are embedded length fields, following the Count Field, before each SMID Field. Within each SMID i, there may or may not be embedded length fields depending on its Control Field.

1) X'B0' X'03' X'08' X'A1' MACKEY3 X'02' X'A1' 2 X'05' X'A1' ACME

— The Count Field indicates 3 SMIDs follow.

— The first is 8 bytes long and identifies key name MACKEY3 to be used with the Fixed Transaction Key method.

— The second is 2 bytes long and identifies key name 2 to be used with the Fixed Transaction Key method.

— The third is 5 bytes long and identifies key name ACME to be used with the Fixed Transaction Key method.

2) X'B0' X'02' X'08' X'A1' MACKEY3 X'03' X'A2' X'14A3'

— The Count Field indicates 2 SMIDs follow.

— The first is 8 bytes long and identifies key name MACKEY3 to be used with the Fixed Transaction Key method.

— The second is 3 bytes long and identifies key name X'14A3' to be used with the Master Key/Transaction Key method.

3) X'B0' X'02' X'27' X'C2' CSM(MCL/RSM RCV/A ORG/B MAC/FEDC BA98)
X'47' X'C2' CSM(MCL/KSM RCV/A ORG/B
KD/1122334455667788.P.3 CTP/002 MAC/FFEE DDCC)

— The Count Field indicates 2 SMIDs follow.

— The first is 27 hexadecimal bytes (39 decimal) long and conveys a CSM from node B acknowledging receipt of the last key received from node A.

— The second is 47 hexadecimal bytes (71 decimal) long and conveys a CSM from node B to distribute key 3, encrypted, to node A.

# Annex C
# (Informative)
# Initial Key Distribution

(This Annex is not part of this standard and is included for information only.)

## C.1 Overview of Key Management

The following is the process of initial key distribution in the retail banking environment. In order to securely process PINs in this environment, keying relationships are established during the initial key distribution process. To better understand the purpose and intent of initial keys and keying relationships,

**Figure C-1**.is an example transaction flow, starting at the PIN Entry Process where the PIN is entered, to an intermediate PIN Translation Process (Acquirer / Switch), to the PIN Verification Process (Issuer). In each of these processes, functions performed on PINs and /or keys must occur within a cryptographic device (TRSM).

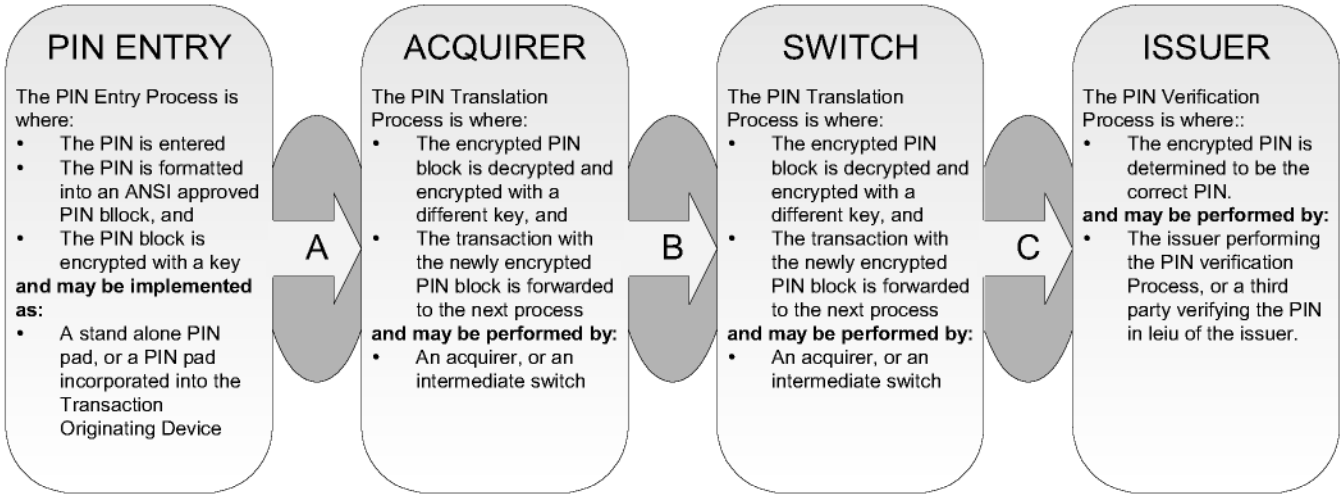



**Figure C-1 - Example transaction flow**

Regardless of the physical implementation, the keying relationships that exist between the communicating parties are really a function of the processes and their deployment. With regard to the initial DEA key distribution, which established the keying relationship, the above process can be generically addressed as the PIN Entry Point, the Transaction Sender, and the Transaction Receiver.

Figure C-1 also shows three initial key distribution relationships between:

A. the PIN Entry Point and the Acquirer,

B. the Acquirer and the Switch, and

C. the Switch and the Issuer.

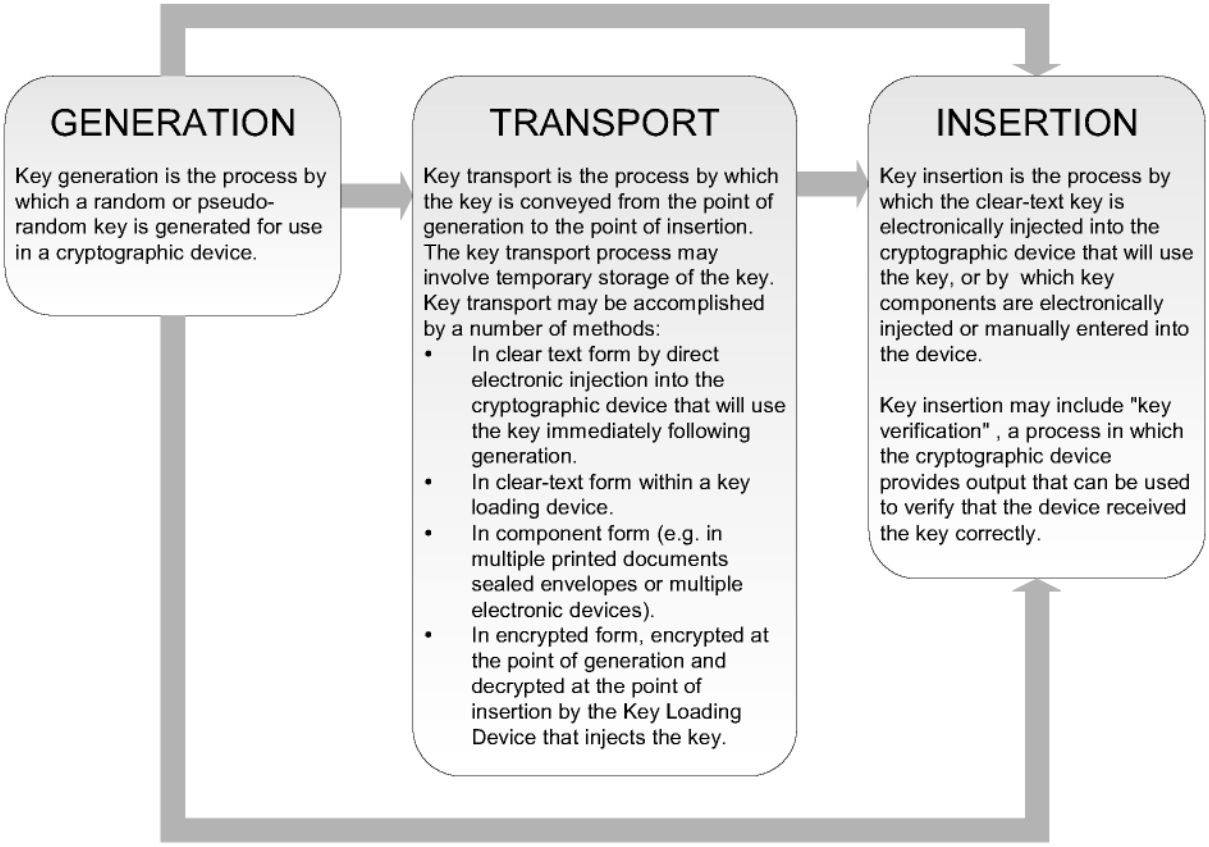



**Figure C-2 - Characteristics of initial key distribution**

**Figure C-2** shows the initial key distribution steps, key generation, key transport, and key insertion. The three paths delineated by arrows depict the possible techniques. The initial DEA key in any cryptographic device is a clear-text key. For any symmetric cipher (e.g. TDEA) the initial key is the first key and no prior keying relationship exists. Therefore, the initial key distribution process needs to be accomplished in a secure manner. The initial key distribution process is categorized into three steps,

- The top path is where the initial key is generated and used inside a key generation device.
- The bottom path is where the initial key is generated inside a key generation device and directly injected into the destination device which is physically connected to the key generation device.
- The middle path is where the following steps occur sequentially:

  a) the initial key is generated inside a key generation device and transferred to an intermediate transport device or split into components,

  b) the initial key is transported to a different physical location,

  c) the initial key is inserted into the destination device.

For each step (generation, transport, and insertion), there are several techniques available to be used, security requirements to be met, and business decisions to be made, that increase or reduce probable risk factors. This annex addresses each of these topics.

## C.2 Objectives of initial key distribution

The objectives of initial key distribution are to ensure that, before a cryptographic device is placed in service:

- the device contains a secret key that no person is able to ascertain, and
- the device is genuine, and has not been subject to unauthorized substitution or modification.

The need for key secrecy is obvious and well understood. However the need to prevent unauthorized substitution or modification is also very important (see Section 7.1, so that an adversary cannot make physical or functional alterations that compromise security (e.g. result in PIN disclosure).

As an example, PIN usage can require that the PIN and the magnetic stripe data, all the information that is necessary to produce counterfeit cards usable (for example) for fraudulent ATM withdrawals, be entrusted to equipment located in a non-secure retail environment. The magnetic stripe data could easily be obtained in this environment. To ascertain PINs, adversaries might attempt, prior to key insertion:

(1) modify a PIN entry device by placing within it a “bug” to disclose PINs during the PIN-entry process before PIN encryption, or

(2) substitute for the legitimate PIN entry device a counterfeit “look-alike” device having all the correct operational capabilities plus fraudulent capabilities as well, such as a PIN-disclosing “bug”. (The “counterfeit” device might be a once legitimate unit that had been stolen from another location and fraudulently altered.)

Thus it is important to ensure the legitimacy of the PIN entry devices that are placed in service.

The process of initial key distribution can achieve both of these objectives, key secrecy and assurance of cryptographic-device legitimacy. It can ensure that the initially loaded key is secret, unavailable to any individual. The process can also confirm the legitimacy of the cryptographic device, “certifying” in a subsequently verifiable way that the device is legitimate and has not been subject to unauthorized substitution or modification.

## C.3 Requirements for initial key distribution

Requirements for each of the key-distribution steps are as follows:

### C.3.1 Key generation

Keys and key components are generated by a random or pseudo-random process such that it is not feasible to determine any key or to determine that certain keys are more probable than other keys from the set of all possible keys. (See Section 7.4)

The initial key (and all subsequent keys) are generated to be unique (except by chance) for each cryptographic device, and the disclosure of the key in one such device must provide no information that could be feasibly used to determine the key in any other such device. (See section 7.1 and 7.6)

For example:

- When using the fixed key technique or master-key/session-key technique, all keys are unique (except by chance) to each cryptographic device. (See section 7.6)
- When keys are generated by a derivation process, this process must ensure that all cryptographic device keys that are derived from the same Base Key (i.e. Derivation Key) use unique data (e.g. a unique “initial key serial number”) for the derivation process, so that all such cryptographic devices receive unique keys. (See section 7.6)

A key for a cryptographic device should be generated electronically within a TRSM meeting the requirements of section 7.2. If the key-generating TRSM does not retain any information that could disclose any previously generated key that has been loaded into a cryptographic device, the requirements for a "unique key per transaction" TRSM apply. If the TRSM does retain such information, the requirements for a "physical barriers" TRSM are used. (See section 7.2)

No cleat-text cryptographic key may be in a general purpose computer, unless this computer meets the requirements for a TRSM. (See section 7.2)

If keys are not generated within a TRSM, they are generated as components. (See section 7.1)

The key generation process:

- Never outputs more than one copy of a clear-text key that is to be directly transferred into a cryptographic device.
- May output any number of copies of this key in encrypted form.
- Outputs no more than the minimum number of sets of components consistent with effective system operation. (If multiple sets of components are required for the same key, then different key components may be used for each set.).

Compromise of the key generation process is not possible without collusion between two trusted people.

The output of the key generation process is monitored by at least two trusted people, who ensure that there is no unauthorized "tap" or other unauthorized means that might disclose a clear-text key or key component as it is transferred between the key generating TRSM and the device or medium receiving the key or component.

Printed key components are printed within blind mailers, or sealed immediately after printing, so that each component can be observed only by the party trusted with it, and so that tampering can be detected.

Key generation are performed in a secure facility that physically and procedurally protects against disclosure all keying material existing within the facility. (See section 7.3)

### C.3.2 Key transport

No person may have access to any clear-text key during the transport process.

When a key is transported as components it is conveyed as at least two full-length components. (See section 7.1)

There are no special requirements for keys transported in encrypted form.

The probability of collusion is reduced if a minimum number of people (consistent with effective operation of the system) have access to any key component or to the medium conveying this component. A person with access to one component of a key, or to the medium conveying this component, must not have access to any other component of this key or to any medium conveying such other component. (See section 7.5)

Organizationally, care should be made to choose Key Custodians so there is limited chance of collusion. For example, close family members, supervisor / subordinate relationships, etc. Procedures should be written to reduce any chance of collusion or the perception that collusion might be possible.

Procedures are implemented to prevent or detect any disclosure of a transported key or key component. (See section 7.1, 7.2 and 7.5)

If disclosure of a key or key component is known or suspected, provision are made so that this compromised key cannot be substituted for a legitimate key. (See section 7.1 and 7.2)

A Key Loading Device that contains a clear-text key are a TRSM, designed and implemented in such a way that any unauthorized disclosure of the key is prevented or detected. (See sections 7.1 and 7.2)

Any Key Loading Device containing a clear-text key must, at all times, be under the continuous supervision of a trusted person, or locked in a security container that is constructed so as to detect unauthorized access. (See section 7.2)

Any document or medium containing a clear-text key component is at all times:

- under the continuous supervision of a trusted person,
- locked in a security container meeting the above requirements, or
- in transit by secure means. A key-component document or medium is in transit by secure means if it is sent via:
    - secure courier (e.g. registered mail) in tamper-evident authenticable packaging,
    - the components of a key are sent at different times so that it is unlikely that personnel involved in the transit process would be able to correlate them.
    - The packaging should be marked with a unique number, for validation by the sender, This eliminates the chance that the component has been accessed and repackaged in a similar package.

A printed key component may reside only within a tamper-evident and non-transparent sealed envelope such that the component cannot be ascertained without opening the envelope.

A key component are printed or recorded in such a way that no unauthorized person (e.g. a person with access to another component of this same key) can ascertain it. Any residue from the printing or recording process that might disclose the component are destroyed before it can be obtained by an unauthorized person. (See section 7.5)

### C.3.3 Key insertion

A key may be inserted into a cryptographic device only when there is a high degree of assurance that the device has not been subject to unauthorized substitution or unauthorized modification. This requires either physical protection of the device against unauthorized access during manufacturing and up to the point of key insertion, or inspection, and possibly testing, of the device immediately prior to key insertion.

The key insertion process is performed under dual control. This is to ensure that the key is being inserted into the intended device, and that there is no “tap” or other means of disclosure of the key or key component between the conveyance medium and the cryptographic device. Such dual control can be enforced by access-control mechanisms.

The key insertion process ensures that each cryptographic device is given keys that are unique and that are not related (except by chance) to any key in any other such device. After key components are loaded for the first time, the components are securely stored. If key components are not required for possible subsequent use, they are destroyed. Re-insertion of the same key components into a cryptographic device is permitted only when there is no suspicion that either the originally loaded key or the device has been compromised. After a clear-text key has been inserted into a cryptographic device from a Key Loading Device, the Key Loading Device are securely stored.

If the key is not required for possible subsequent re-insertion, it should be erased from the Key Loading Device. (See Section 7.5.2)

If attempts to insert a key or key component into a cryptographic device all fail, the same key or component may not be loaded into a replacement device unless it can be ensured that all residue of the key or component has been erased or otherwise destroyed in the original device. Some PEDS may not have an “erase key” function: a method for key erasure is to inject another SECRET key to replace the key of interest.

#### C.3.3.1 Specific requirements for printed key components

The key-component document must not be opened until the time when the component is to be entered into a cryptographic device. (See section 7.5) The envelopes containing the component are clearly marked to indicate the contents without requiring that the envelope be opened, and this is best done with the person entrusted with a key-component document, immediately prior to opening the document, closely inspects it for signs of tampering. If tampering is observed or suspected, the component and the associated key are not used and the associated key is erased or otherwise invalidated at all locations where it is known to exist. (This is to prevent the use or substitution of the compromised key in place of a legitimate key.)

Only the person entrusted with a key-component document is authorized to open the document, and this person must ensure that no other person can observe the printed key component. This person must then enter the component into the cryptographic device in such a way that no other person can observe the entered value. (See section 7.5)

If the component is to be stored for possible reuse, it is stored so as to prevent its unauthorized disclosure.

#### C.3.3.2 Specific requirements for key components transported in an electronic medium

A person entrusted with a component must initiate and oversee the process by which this component is injected from the electronic medium into the cryptographic device. (See section 7.5)

Once the component injection is completed, and correct receipt of the component is confirmed (if applicable), then either:

- the medium are placed in secure storage, if it may be required for future re-insertion of the key into the cryptographic device (see section 7.5.2), or
- all trace of the component are erased or otherwise destroyed from the electronic medium. (For additional information about erasure, see ANSI X9.8, Annex F)

#### C.3.3.3 Specific requirements for keys transported in encrypted form

After successful injection of the decrypted key, erasure from the Key Loading Device is performed.

Immediately after the decryption, the key-encrypting key is erased if it is no longer needed.

## C.4 Implementation considerations

The following provides implementation guidance for each of the steps in initial key distribution, and for the control of cryptographic devices both prior to and subsequent to initial key distribution.

### C.4.1 Key generation

As stated in the requirements, use of an initial key-generating TRSM is under dual control. It may also be necessary to ensure that access to the TRSM is under dual control.

Some implementation examples for key generation facilities are described in Section C.5.

### C.4.2 Key transport

If the compromise of a key or key component is known or suspected, the key is erased or otherwise invalidated at all places where it legitimately exists,

#### C.4.2.1 Key loading device for cleat-text keys

As stated in Section C.2, a clear-text key is transported in a TRSM. The TRSM itself does not have to be under dual control if it cannot output the key except under dual control, and if it is carefully supervised at all times to prevent it from being accessed by unauthorized personnel. For example this can be achieved by:

- transporting the device in a locked or sealed security container that the person responsible for transport cannot alone open without detection, and recording, and retaining a record of, each instance when the container was opened; or

- ensuring that the TRSM will not output a key without the immediately-prior entry of a password, physical key, or access card that is unavailable to the person responsible for transport.

With either method, each access to the device should be recorded.

#### C.4.2.2 Key components

The following are examples of how key components can be packaged for transport:

- within tamper-evident packaging that is closely inspected prior to use to ensure that the packaging has not been previously opened or substituted;

- within a TRSM.

Security is enhanced if components of a key are sent from different points of origin, so that it is unlikely that personnel involved in the transfer process would be able to correlate them.

#### C.4.2.3 Encrypted keys

Encrypted keys may be transported by any means.

### C.4.3 Key loading

There are two basic techniques to insert a key into a cryptographic device.

- Bring the device to the key. This requires a Key Injection Facility to which the cryptographic device is physically transported. After key loading the device is then transported to its operational location.

- Bring the key to the device. This requires the use of key components or a Key Loading Device to transfer the key from the point of generation to the cryptographic device, most likely in its operational location.

#### C.4.3.1 Use of a key injection facility

A Key Injection Facility is a centralized location where devices are brought for initial key loading. Such a facility operates under dual control, and the injection path is carefully inspected by both parties for evidence of "bugging". A Key Injection Facility is a possible option when the cryptographic device can feasibly be transported to the facility.

There are possible advantages to the use of a Key Injection Facility. A secure facility that is specialized in key distribution may be able to perform this function more securely than it can be performed otherwise. Furthermore it may be more "cost effective" to ship the cryptographic devices to and from a central facility for key insertion than to send personnel with components or Key Loading Devices to the place where each cryptographic device is located. Finally, it may be more secure to deliver a device to a non-secure operational location with the key(s) already installed, as the operationally verified presence of a legitimate key can provide assurance that the device was not subject to unauthorized modification or substitution during the shipping, storage, and installation that precede actual use. Thus the use of such a facility can simplify compliance with the requirement (section 7.2) that a key may be loaded into a cryptographic device only when it can be assured that the device has not been subject to substitution or to physical or functional modification. The cryptographic device can be shipped directly from the manufacturer to the facility by relatively secure means, so that it can reasonably be ensured that the device was not subject to substitution or modification prior to initial key distribution.

An example of key loading at a manufacturer's facility is described in C.6.

A Key Injection Facility may also serve as a Key Generation Facility.

#### C.4.3.2 Use of key components or key loading devices

Under some conditions it is not feasible to transport the cryptographic device to a Key Injection Facility, and the key has to be transported to the cryptographic device either as components or in a Key Loading Device. When this is done, precautions are needed to ensure compliance with the following:

- the cryptographic device has not been subject to unauthorized substitution or modification prior to initial key insertion;
- the path that the key or component will traverse to enter the cryptographic device has not been "tapped";
- the proper conditions of dual control and security of the environment are present at the time of key insertion.

### C.4.4 Protection of cryptographic devices

The requirements for initial key distribution (Section C.3) necessitate protection for a cryptographic device before and after it has received its initial key. If features of the device itself do not provide adequate protection against fraudulent modification or substitution, the manufacturers and deployers of cryptographic devices have to exercise appropriate controls during and subsequent to manufacture.

#### C.4.4.1 Protection prior to key insertion

The requirements necessitate a high degree of protection for the cryptographic device before its initial key has been inserted. In its pre-loaded state, the device may be susceptible to fraudulent modification (e.g. installing a PIN-disclosing "bug") or substitution (e.g. replacing the unit with a counterfeit device having the correct visual and operational characteristics in addition to fraudulent capabilities such as a PIN-disclosing "bug".) It is therefore important that manufacturers and deployers of cryptographic devices exercise appropriate control over the units during and subsequent to manufacture. Although a high degree of physical control over cryptographic devices may be required prior to key insertion, reasonable accountability is important subsequent to key insertion. Such accountability is important so that if a device is lost, stolen, or appears to have been tampered with, and as a result

a security compromise is suspected, the key loaded into the device can be identified in order to detect any attempted use of this key.

It is also important that the units are conveyed to and stored at the place of key insertion so as to ensure that the units are not subject to unauthorized modification or substitution. Four possible ways of achieving this are as follows:

1 The cryptographic devices are transported from the manufacturer's facility to the place of key insertion using a trusted courier service (e.g. Registered Mail), and then are securely stored at this location until key insertion occurs.

2 The cryptographic devices are shipped from the manufacturer's facility to the place of key insertion in serialized, counterfeit-resistant, tamper-evident packaging, and then are stored in such packaging, or in secure storage, until key insertion occurs.

3 The manufacturer's facility loads into each cryptographic device a secret, device-unique "transport-protection token". The TRSM used for key injection has the capability to verify the presence of the correct "transport-protection token" before overwriting this value with the initial key that will be used, for example, PIN encryption.

4 Each cryptographic device is carefully inspected and perhaps tested immediately prior to key insertion, using due diligence, to provide reasonable assurance that it is the legitimate device and that it has not been subject to any unauthorized modifications.

#### C.4.4.2 Protection subsequent to key insertion

Although a high degree of physical control over cryptographic devices may be required prior to key insertion, reasonable accountability is important subsequent to key insertion. Such accountability is important so that if a device is lost, stolen, or appears to have been tampered with, and as a result a security compromise is suspected, the key loaded into the device can be identified in order to detect any attempted use of this key.

To simplify such accountability, it is desirable to have some means of identifying the just-injected key. The manufacturer's device serial number can be the basis for accountability, provided records are maintained that cross-references this serial number to the identity of the key. To eliminate the need for such cross-referencing, it may (in some environments) to convenient to print on a label the "key name" of the device's key and affix this label to the device. This key identification then becomes the basis for the cryptographic device's accountability.

When a label giving the Key Name is attached to the cryptographic device, this label can be used to ensure that the cryptographic device is not delivered to the wrong destination. Before the cryptographic device is delivered, the Key Name can be checked to ensure that it is correct for the intended destination.

In some environments it may be advantageous if the key identification information can be electronically read from the cryptographic device instead of or in addition to being visually read from a label attached to the device.

As a part of the accountability process it is recommended that a Key Injection Facility retain a record of the identify of every cryptographic-device key it loads, along with the date of loading.

Control of cryptographic devices is necessary for both logistical and security considerations. When a Key Injection Facility is used to load the device's initial key, months or even years may elapse before this device will be installed and it is often not possible to determine at key-loading time when and where the device will be placed in service. This may cause logistical problems for the acquirer, which, when a cryptographic device is installed, needs to determine the key initially loaded into this device (perhaps years earlier). To solve these problems and identify the device's key, the acquirer can either relate the key's identity to the identity of a hardware device (e.g. the PIN pad or terminal) that the acquirer can determine, or use "key identification by name" in which a non-secret key identifier is loaded into the device along with the initial key, and transmitted in a Security Management Information Data

Element (SMID), from they cryptographic device to the acquirer with every secured transaction. See Annex B for a discussion of the SMID and how it can be used to determine the key in a cryptographic device.

Another method of providing accountability, applicable to installed cryptographic device, is through a "challenge/response" process. A host system transmits a challenge to the device, which responds with a cryptographically authenticated response to the challenge. The host system, upon verifying the cryptographically authenticated response, has a high degree of assurance that the device in question is till in its operational location.

If accountability procedures determine that a cryptographic device is lost or stolen the identify of the key in the missing device is given to the appropriate acquirer, and the acquirer then ensures that a cryptographic device using this key is detected and investigated if an attempt is ever made to place it into service.

### C.4.5 Reloading of cryptographic devices

In some situations it may be necessary to load a different key into a cryptographic device, for example, when the merchant owning the device changes from one acquirer to another. The reloading process is a repeat of the key-injection process. Care are taken to provide reasonable assurance that the cryptographic device is not subject to unauthorized modification or substitution subsequent to its last use with the "old" key and prior to the injection of a "new" key. The following are some examples of how this can be accomplished:

- Physically protecting the device against access by unauthorized people from the time the device is last used with the "old" key until the "new" key injected.
- Verifying the presence of the "old" key in the device immediately before injecting the "new" key.
- Inspecting/testing the device immediately prior to loading the "new" key to provide reasonable assurance that the device is legitimate and has not been subject to unauthorized modification.

## C.5 Example of manual key distribution

This is an example of manual key distribution using key components under dual control and split knowledge. This example is applicable to both DEA and TDEA. For DEA, keys and key components are 56 bits plus 8 parity bits; for TDEA, keys and key components are at least 112 bits plus 16 parity bits.

Once a key (or its key components) is generated using dual control, the key is then distributed by the following steps:

— 'recording' the key component digits as they are generated or recording the key in a TRSM as it is generated.

— 'transporting' or securely mailing the recorded key or key component digits to both TRSMs receiving the key

— 'loading' the recorded key or its key component digits into the TRSM.

Under dual control procedures, the key is conveyed as 'n' key components with the same structure as the generated key. As the name 'dual control' implies, the number ('n') of key components will be two or more. The XOR operation is used to combine the key components and produce the actual key. The XOR operation is performed in a pair-wise fashion on the key components in any sequence. The process may be viewed from either of two perspectives:

a) Key is generated by combining generated key components (i.e., the perspective taken when manually generating a key)

1) Generate 'n' (pseudo-)random key components

2) (Optional) Provide correct parity to each key component

3) (Optional) Compute check value for each of 'n' key components

4) (Optional) Derive the resulting key by XOR of the key components

5) (Optional) Provide correct parity to the resulting key

6) (Optional) Compute check value for the resulting key

b) Key is generated prior to creating key components

1) Generate (pseudo-)random key

2) (Optional) Provide correct parity for the key

3) (Optional) Compute check value for key

4) Generate 'n-l' (pseudo-)random key components

5) Derive the final 'n' key component by XORing of the key (from step a) and each of the generated 'n-l' key components

6) (Optional) Provide correct parity to each of 'n' key components

7) (Optional) Compute check value for each of the 'n' key components

| Component or XOR Product | Hexadecimal value (16 Hex digits) | Binary Equivalent (64 Bits) [1] | Check value [2,3] |
|---|---|---|---|
| Component 1 | 5B0E B037 43F1 5D9D | 01011011...10011101 | 82157A |
| **XOR** | **XOR** | **XOR** | |
| Component 2 | F832 7F94 043B 8C85 | 11111000...10000101 | 733C48 |
| Temp. Result | A33C CFA3 47CA D118 | 10100011...00011000 | N/A |
| **XOR** | **XOR** | **XOR** | |
| Component 3 | 37BC 92E6 DC5D 254F | 00110111...01001111 | 8C49EC |
| Resulting Key | 9480 5D45 9B97 F457 | 10010100...01010111 | 9B1BBB |

**Table C-1 – Example of Pair-wise XOR Combination of Key components for DEA**

Notes:

1 For illustrative purposes, only the first and last eight bits have been shown for the 64-bit 'Binary Equivalent'.

2 In this illustration the check value is the left-most six hexadecimal digits from the ciphertext produced by using the DEA in ECB mode to encrypt a 64-bit binary zero value with the subject key or key component.

3 For an example of computing Key Check Value for double length keys, use the technique shown in Figure C-1.

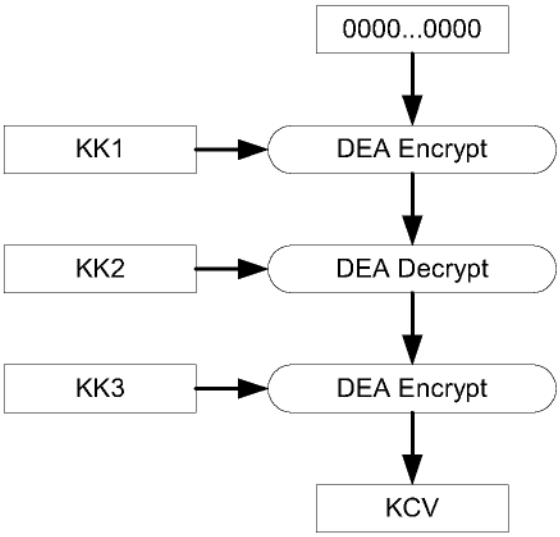



**Figure C-3 – Generating Key Check Value**

The optional check values, as mentioned in notes 2 and 3 above, are the left-most six hexadecimal digits from the ciphertext produced by using the DEA in ECB mode to encrypt to 64-bit binary zero value with the subject key or key component. The check value process may be simplified operationally, while still retaining reliability, by limiting the check value to the left-most four or six hexadecimal digits of the ciphertext. (Using the truncated check value may provide additional security in that the ciphertext which could be used for exhaustive key determination would be unavailable.)

Recording of key components, performed under dual control, generally results in each key component being either printed or written. Optionally, a key component check value is also printed or written. (The key component check value provides a means for verifying the correct receipt and entry of the key component value at the TRSM site during the loading process.) Each key component may be separately printed or securely displayed to permit writing the value, prior to being separately sealed (along with its optional key component check value) in a conveyance envelope. Optionally, a key check value may be included in one or more of the sealed envelopes. (The key check value provides a means for verifying the correct receipt and entry of all key component values at the TRSM site during the loading process.) When either printing or displaying the key component value, it is suggested that the hexadecimal digits appear in either pairs or groups of four to reduce errors in writing and/or later loading.

Under dual control and split knowledge, each key component is confidentially released to one and only one of the 'n' trusted entities.

Transporting of key components occurs by separately transporting the sealed conveyance envelopes to the TRSM site where the key will be used. Each trusted individual transports one key component to the (TRSM) point of key entry. Alternatively, each trusted individual at the site of key generation (e.g., at a Key Initialization Facility) may securely mail (e.g., by bonded courier or registered mail) a confidential key component to one of a parallel set of 'n' trusted individuals at the TRSM site. In this case, procedures for tamper checking of envelopes, proper receipts, and signature verification or other means of authentication are also needed before loading may commence.

Loading starts as each of the 'n' trusted individuals securely and separately enters her/his own key component by using the implemented key entry facilities at the TRSM site. Optionally, the parity and/or check value of the key component are verified at the completion of each key component entry. As each of the key components is entered (and optionally verified), it is added within the TRSM using the XOR operation to the previously entered key components (actually, to the result of the previous XOR operation). Optionally, the TRSM provides correct parity for the subject DEA or TDEA key. Optionally, the key check value is verified to validate the construction of the subject key. Loading, and thus key distribution, is completed.

## C.6 Example of key loading controls at a manufacturer's facility

This example of an initial key loading implementation which meets, and in some cases exceeds, the minimum requirements of this standard. It outlines the unique challenges involved in providing adequate key management at a manufacturing facility which is initializing cryptographic devices. It also provides a description of the environment and the logistical realities that need to be considered by a manufacturing operation when it contracts to use or store a Key Loading Device (KLD).

This annex describes only one way to implement key loading controls that meet the requirements of this standard. There are many other implementations and selecting one depends on many factors including type of manufacturer, size, location, frequency of key loading, etc.

The type of key management used with the devices when in operation does not affect the implementation. And, for the purposes of this example, it is assumed the keys are not generated at the manufacturer's facility. It also assumes the keys and the KLD have been transported to the facility in accordance with the requirements of this standard.

There are a number of factors which shape the use of a KLD on the premises of an equipment manufacturer, including the following.

- Cryptographic devices manufactured at the facility are not necessarily sold to an acquirer, but instead are sold to a third party card acceptor who has established relationships with one or a number of acquirers.
- Cryptographic devices purchased by a third party card acceptor, but manufactured by other suppliers, may be initialized at this facility using the same KLD.
- Key initialization may be a sporadic activity which represents a tiny portion of the manufacturing facility's activities. This may preclude the use of dedicated personnel, floor space or equipment for this activity.
- Vaults or other types of physically secure rooms that can be dedicated to this function may not be available on the premises and may have to be constructed.
- Security has long been a requirement of the manufacturing segment. Access security (both external and internal), materials security, and personnel security are usually standard elements in any manufacturing operation and may only need to be enhanced to recognize the unique requirements of key loading.

A manufacturing facility doing key loading controls entry to and exit from its premises. No access to the facility for employees or visitors is permitted without authorization. Employee badges containing their photos are visible to security and supervisory personnel at all times. Badges are coded (e.g., colors or shape) to control access to internal areas with higher security requirements, such as the key loading area.

All visitors are "registered" upon entry to the facility and are escorted at all times by an employee.

A professional security service is retained to enforce the above access controls. No property (equipment, cartons, etc.) leaves the facility without a properly authorized property pass.

Published security procedures exist which detail the appropriate response to unauthorized personnel encountered on the premises and provide for areas of the facility with tighter security requirements (e.g., key loading area). Routine security audits of adherence to these procedures are conducted periodically, preferably on a random schedule.

Procedures provide for dual control to be utilized in handling the KLD. Access controls for the KLD are split between two trusted individuals. Selection of the individuals takes into consideration their job history

# Annex D
# (Informative)
# Key Set Identifiers

(This Annex is not part of this standard and is included for information only.)

As long as transactions from a given set of terminal TRSMs will go to only one acquirer, or to any of a predetermined group of acquirers, a standardized technique for assigning Key Set Identifiers is not required. The acquirer, or the group of acquirers, may assign these numbers in any desired way, ensuring only that each Key Set Identifier is unique. However if transactions from such a TRSM set may at some future time go to an acquirer not originally anticipated, use of a standardized method for assigning Key Set Identifiers is recommended. Otherwise there is the possibility that the Key Set Identifier associated with the TRSM set in question is, by chance, identical to a Key Set Identifier already in use by this new acquirer for another set of TRSMs.

Virtually all financial institutions issue cards, and therefore have been assigned one or more six-digit issuer identification numbers per Reference 6. Any such number is unique to the institution to which it is issued. Therefore the recommended technique for assigning Key Set Identifiers utilizes these already-assigned issuer identification numbers. (See IS 7812 and/or IS 8583.)

An institution may use an issuer identification number as a Key Set Identifier, provided it will never need more Key Set Identifiers than the quantity of issuer identification numbers it has been assigned. If the institution may need additional Key Set Identifiers, it may concatenate one or more hex digits to the right of an issuer identification number, and in this way obtain 16, 256, or more Key Set Identifiers from a single such number.

Key Set Identifiers do not have any specified maximum length, and the acquirer stores with each Key Set Identifier the length of (the number of hex digits in) that identifier. When the acquirer receives a transaction from a terminal TRSM, the acquirer attempts to match each of its Key Set Identifiers with as many SMID digits as are specified for that particular Key Set Identifier. Such a match indicates that the Key Set Identifier in question is the one which applies to this particular SMID.

Because Key Set Identifiers do not have a fixed length, the financial institution which is allocating such identifiers decides how many hex digits (if any) to concatenate with its issuer identification number to obtain its Key Set Identifiers, prior to allocating any Key Set Identifiers based on this issuer identification number. For example if such an institution decides to use seven-hex digit Key Set Identifiers by concatenating a single hex digit with an issuer identification number, it cannot, after having utilized all 16 such seven-hex digit numbers, subsequently add an 8th digit to obtain additional Key Set Identifiers. Such an eight-hex digit Key Set Identifier would match, in the first seven digits, a Key Set Identifier already in use.

An organization which wishes to obtain a Key Set Identifier but which does not issue cards (e.g., a large retailer) may obtain such an identifier from a financial institution which does issue cards. Such a financial institution ensures that it never assigns duplicate Key Set Identifiers.

## D.1 An Example Key Serial Number Format

The Derived Unique Key Per Transaction (DUKPT) key management technique uses a 10 byte Key Serial Number (KSN) to uniquely identify each of the derived keys. The DUKPT technique is described in Section 8.7 of this part of this standard. The standard does not fully specify the format of the 10-byte KSN. The DUKPT technique requires the right-most 21 bits be used as a counter that is increased for each successively derived key. The left two bytes do not participate in the derivation process and serve to identify or partially identify the base derivation key. The right-most 8 bytes are used in the derivation process. Of the right-most 8 bytes of the KSN, the right-most 21 bits are the counter and the use of the remaining 43 bits is not specified, but are unique for each TRSM that uses the same base derivation key.

An example format for the 10-byte KSN is shown in Figure D-1.

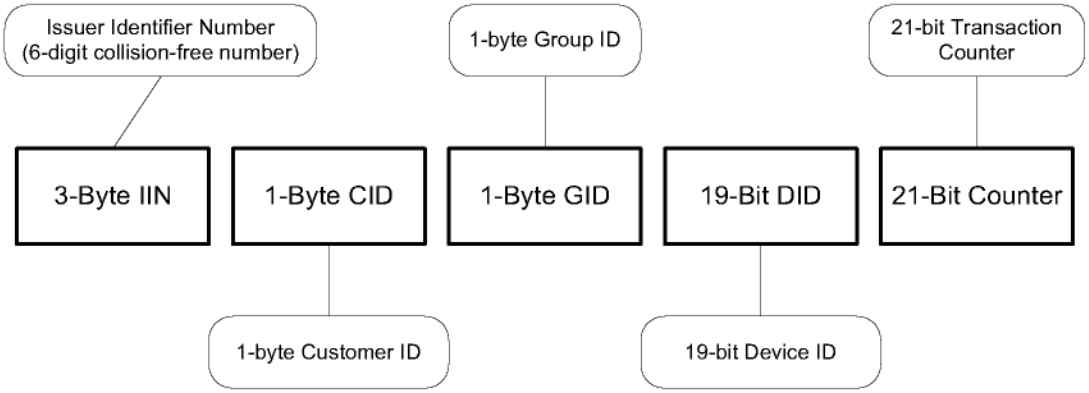



**Figure D-1 – Key Serial Number Format Example**

### D.1.1 IIN - 3 Bytes - Issuer Identification Number

The IIN is a collision free 6 digit number which can be obtained from the ABA for a fee. Use of this number will ensure the uniqueness of the KSN. The IIN is used for the device in much the same way the Bank Identification Number (BIN) is used for a magnetic stripe card.

### D.1.2 CID - 1 Byte - Customer ID

The CID can be used by an acquirer or device issuer to segregate customers from each other. For example, an acquirer might have a chain of hardware stores (e.g., Smith's Hardware Stores), and perhaps a chain of clothing stores (e.g., Wide Body Men's shops). The acquirer might assign CID=14 to devices issued to Wide Body Men's Shops and CID=26 to devices deployed to Smith's Hardware Stores. The CID can provide a method to quickly and easily determine to which customer a device had been issued.

### D.1.3 GID - 1 Byte - Group ID

The GID can be used by an acquirer or device issuer to separate devices for a given Customer. The GID can be assigned to some arbitrary values that have meaning to the acquirer or device issuer. For example the GID could be used to identify a particular device vendor or model of device. The GID could also be used to indicate the time period in which the device was issued. If the GID is used as a date indicator, it could take on values of 00-99 to represent a month or quarter after some date selected by the device issuer. For example if January 1997 is assigned the value of 01, March 1997 might be 03. If hexadecimal values were acceptable, the coding might be: month = 1-C, year = 0 - F. This scheme would provide up to 15 years by month for device separation.

### D.1.4 DID - 19 Bit Device ID

The right-most 5 bytes of the KSN is divided into two fields. The left-most 19 bits constitute the Device ID field and the right-most 21 bits constitute the Transaction Counter field described below. The DID can be used to designate an individual device identification within a specific GID. The easiest way to manage the DID is to use the left 16 bits but ignore the high order 3 bits in the middle byte. For systems using only decimal numbers, such a strategy would provide for 9999 devices within a specific GID. The three high order bits of the middle byte do not participate in the key derivation process and are passed along unaltered. They are therefore available for any user defined function.

### D.1.5 TCTR - 21 Bit Transaction Counter

The right-most 21 bits of the KSN define a transaction counter under the control of the secure DUKPT firmware. The value of the counter is controlled by the firmware and changes to the value of the counter as supplied by the firmware will produce inconsistent and meaningless results. Using the Counter value for the detection of message replay is possible but not defined in this part of this standard.